Lab Manual

Data Warehousing and Mining Lab

CSL503

Branch – Computer Engineering

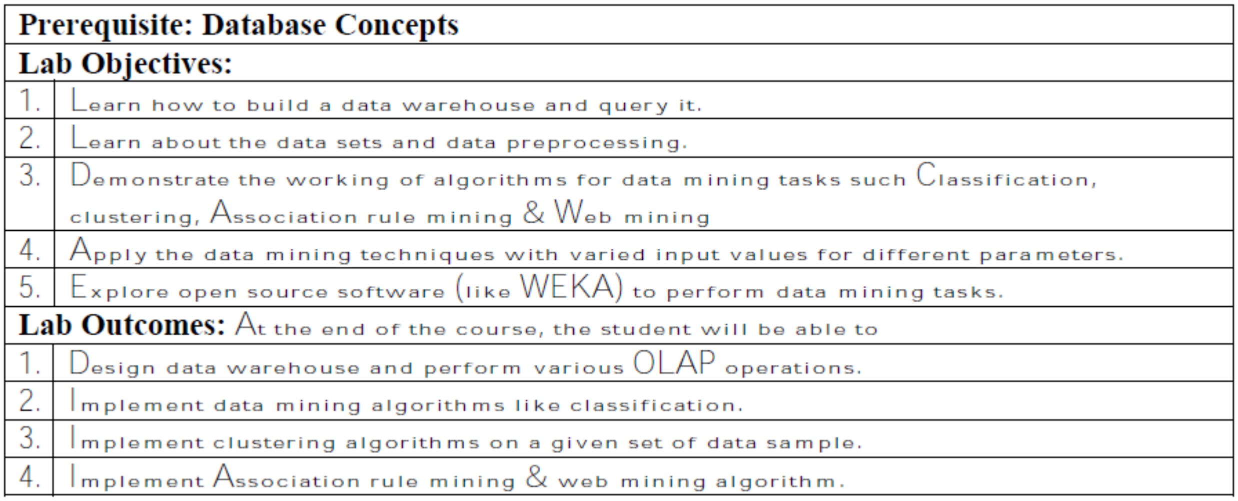
Semester – V

Subject Incharge – Dr Anil Kale

List of Experiments

Sr. No.	Experiments Name
1	One case study on building Data warehouse System / Data Mart. Write Detail Statement Problem and creation of dimensional modelli (creation star and snowflake schema)
2	Implementation of all dimension table and fact table based on experiment case study
3	Implementation of OLAP operations: Slice, Dice, Rollup, Drilldown and Piv based on experiment 1 case study
4	To study the file formats for the data mining.
5	Create an Employee Table with the help of Data Mining Tool WEKA.
6	Apply Pre-Processing techniques to the training data set of Weather Table
7	To implement like Decision tree, Naïve Bayes, Random Forest using WEKA
8	Implementation of Data Discretization (any one) & Visualization (any one)
9	To implement the following Clustering Algorithms – K-means, Agglomerativ Divisive using WEKA
10	Implementation of Apriori algorithm in WEKA.

Course Objectives & Course Outcome, Experiment Plan



Prerequisite: Database Concepts	
Lab Objectives:	
1.	Learn how to build a data warehouse and query it.
2.	Learn about the data sets and data preprocessing.
3.	Demonstrate the working of algorithms for data mining tasks such Classification, clustering, Association rule mining & Web mining
4.	Apply the data mining techniques with varied input values for different parameters.
5.	Explore open source software (like WEKA) to perform data mining tasks.
Lab Outcomes: At the end of the course, the student will be able to	
1.	Design data warehouse and perform various OLAP operations.
2.	Implement data mining algorithms like classification.
3.	Implement clustering algorithms on a given set of data sample.
4.	Implement Association rule mining & web mining algorithm.

Experiment Plan

Module No.	Week No.	Experiments Name	Course Outcome
1	W1	One case study on building Data Warehouse System / Data Mart. Write Detail Statement Problem and creation of dimensional modelling (creation star and snowflake schema)	CO1
2	W2	Implementation of all dimension table and fact table based on experiment 1 case study	CO5
3	W3	Implementation of OLAP operations: Slice, Dice, Rollup, Drilldown and Pivot based on experiment 1 case study	CO5
4	W4	To study the file formats for the data mining.	CO5
5	W5	Create an Employee Table with the help of Data Mining Tool WEKA.	CO5
6	W6	Apply Pre-Processing techniques to the training data set of Weather Table	CO5
7	W7	To implement like Decision tree, Naïve Bayes, Random Forest using WEKA	CO5
8	W8	Implementation of Data Discretization (any one) & Visualization (any one)	CO3
9	W8	To implement the following Clustering Algorithms – K-means, Agglomerative, Divisive using WEKA	CO4
10	W9	Implementation of Apriori algorithm in WEKA.	CO5

Mapping Course Outcomes (CO) - Program Outcomes (PO)

Subject Weight	**Course Outcomes**		**Contribution to Program outcomes**											
			P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**P1**	**P12**
PRATICAL 50%	CO1	Student will be able to understand data warehouse and design model of data warehouse.	1	1	2					1	2	1	1	1
	CO2	Students will be able to learned steps of pre-processing		3	1		1				1	1	2	1
	CO3	Students will be able to understand the analytical operations on data.		1	1		1				1	1	2	3
	CO4	Students will be able to discover patterns and knowledge from datawarehouse.		1			1			1	1	1	2	3
	CO5	Students will be able to understand and implement classical algorithms in data mining and data warehousing; students will be able to assess the strengths and weaknesses of the algorithms, identify the application area of algorithms, and apply them.		1			1			1	2	1	2	2

Study and Evaluation Scheme

Course Code	Course Name	Teaching Scheme			Credits Assigned			
CPC801	Data Warehou and Mining	Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
		04	02	--	04	01	--	05

Course Code	Course Name	Examination Scheme		
CPC801	Data Warehou and Mining	Term Work	Practical	Total
		25	25	50

Term Work:

Internal Assessment consists of two tests. Test 1, an Institution level central test, isfor 20 marks and is to be based on a minimum of 40% of the syllabus. Test 2 isalso for 20 marks and is to be based on the remaining syllabus. Test 2 may beeither a class test or assignment on live problems or course project

Practical & Oral:

Oral examination is to be conducted by pair of internal and external examiners based on the syllabus.

EXPERIMENT NO. 01

Case study on Data Warehouse System

Aim: One case study on building Data warehouse System / Data Mart.

Write Detail Statement Problem and creation of dimensional modelling (creation star and snowflake schema)

Objectives: From this experiment, the student will be able to

- Understand the basics of Data Warehouse
- Understand the design model of Data Warehouse
- Study methodology of engineering legacy databases for data warehousing

Outcomes: The learner will be able to

- Apply knowledge of legacy databases in creating data warehouse
- Understand, identify, analyse and design the warehouse
- Use current techniques, skills and tools necessary for designing a data warehouse

Software Required: Oracle 11g / MYSQL

Theory:

Data Warehouse

A data warehouse is a large collection of business-related historical data that would be used to make business decisions.

- Data warehouse stores aggregated transactional data, transformed and stored for analytical purposes.
- Data warehouses store data from multiple sources, which makes it easier to analyze.

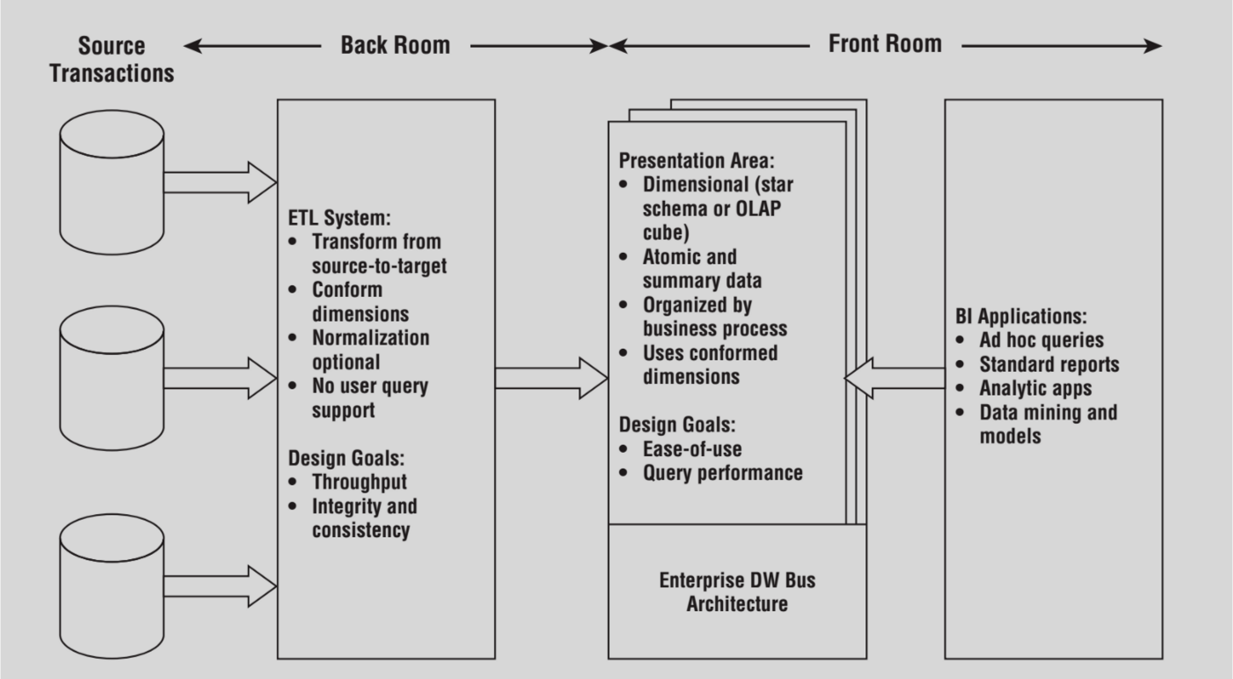
"Simply speaking, the database (operational) systems are where you put the data in, and the Data warehouse (Business Intelligence) system is where you get the data out." — Ralph Kimball

Dimensional Modeling

Dimensional modeling is the widely used technique to design data warehouse mainly because it addresses below two requirements simultaneously:

1. Delivers the data that is understandable by business users.

2. Deliver fast query performance.



Core elements of the Kimball DW/BI architecture

The figure shows the major components involved in building the Data warehouse from operational data sources to analytical tools to support business decisions through ETL (Extract, Transformation, Load) process.

Now let’s take the use case of e-Wallet to build a data warehouse using dimensional modeling technique.

Background

One of the online retail company’s features is an e-wallet service, that holds credit that can be used to pay for products purchased on the platform.

Users can receive credit in three different ways:

1. When a product purchase that is paid for is canceled, the money is refunded as cancellation credit.

2. Users can receive gift card credit as a gift.

3. If a user has a poor service experience, soo-sorry credit may be provided.

Credit in the e-wallet expires after 6 months if it is gift card credit and soo-sorry credit, but in 1 year if it is cancellation credit.

Requirement

The Finance department of the company would like to build reporting and analytics on the e-wallet service so they can understand the extent of the wallet liabilities the company has.

Some of the questions they would want to answer from this are like below:

- What is the daily balance of credit in the e-wallet service?
- How much credit will expire in the next month?
- What is the outcome (i.e. % used, % expired, % left) of credit given in a particular month?

Solution Design

The four key decisions made during the design of a dimensional model include:

1. Select the business process.

2. Declare the grain.

3. Identify the dimensions.

4. Identify the facts.

Let's write down this decision steps for our e-Wallet case:

1. Assumptions: Design is developed based on the *background* (Business Process) given but also keeping flexibility in mind. All the required fields are assumed to be available from the company's transactional database.

2. Grain definition: *Atomic grain refers to the lowest level at which data is captured by a given business process.*

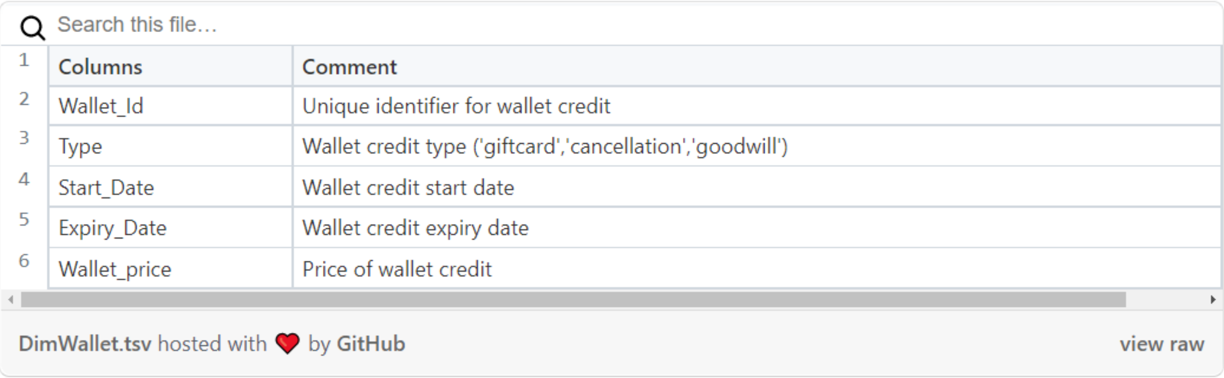
The lowest level of data that can be captured in this context is wallet transactions i.e., all the credit and debit transactions on e-wallet.

3. Dimensions: *Dimensions provide the "who, what, where, when, why, and how" context surrounding a business process event.*

Even though a wide number of descriptive attributes can be added designing dimensions are restricted to the current business process but the model is flexible to add any more details as and when required. (Tables name prefixed with Dim)

Dimension Tables:

- DimWallet

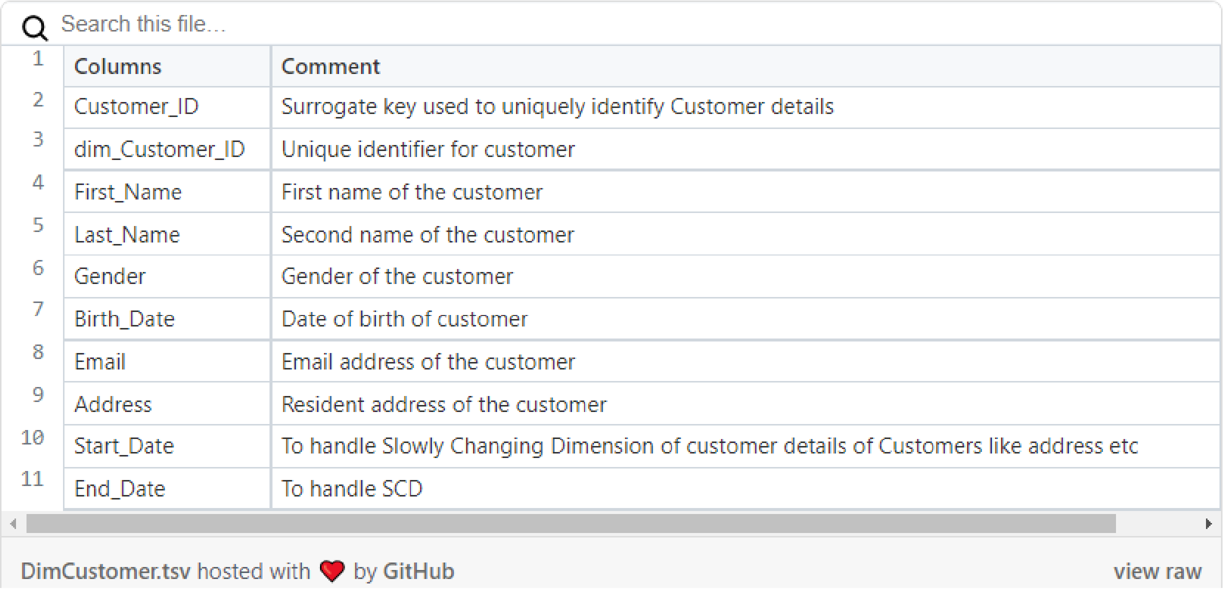


Search this file…

Columns	Comment
Wallet_Id	Unique identifier for wallet credit
Type	Wallet credit type ('giftcard','cancellation','goodwill')
Start_Date	Wallet credit start date
Expiry_Date	Wallet credit expiry date
Wallet_price	Price of wallet credit

DimWallet.tsv hosted with ❤ by GitHub view raw

- DimCustomer



Search this file…

Columns	Comment
Customer_ID	Surrogate key used to uniquely identify Customer details
dim_Customer_ID	Unique identifier for customer
First_Name	First name of the customer
Last_Name	Second name of the customer
Gender	Gender of the customer
Birth_Date	Date of birth of customer
Email	Email address of the customer
Address	Resident address of the customer
Start_Date	To handle Slowly Changing Dimension of customer details of Customers like address etc
End_Date	To handle SCD

DimCustomer.tsv hosted with ❤ by GitHub view raw

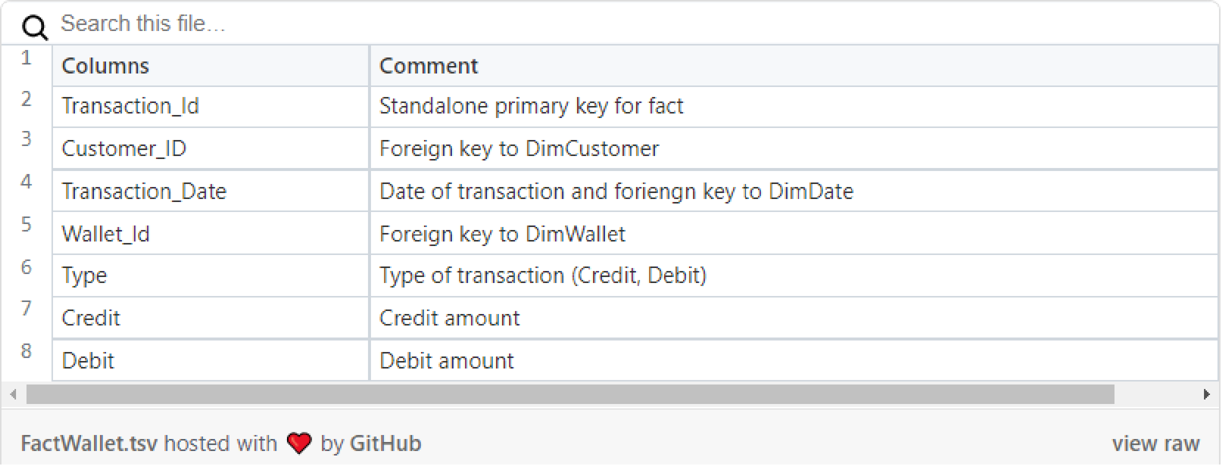
- DimDate: This dimension has all the date related parsed values like Month of the date, Week of the date, Day of the week, etc. This will be very handy to get reports based on time.

4. Facts: *Facts are the measurements that result from a business process event and are almost always numeric.*

Facts are designed such that focusing on having fully additive facts. Even though some business process requirements want facts that are non-additive (% used, % expired, % left, etc). These values can be achieved effectively by calculating the additive facts separately. Each row in fact table represents the physical observable events not only focused on the demands of reports required.

Fact Table:

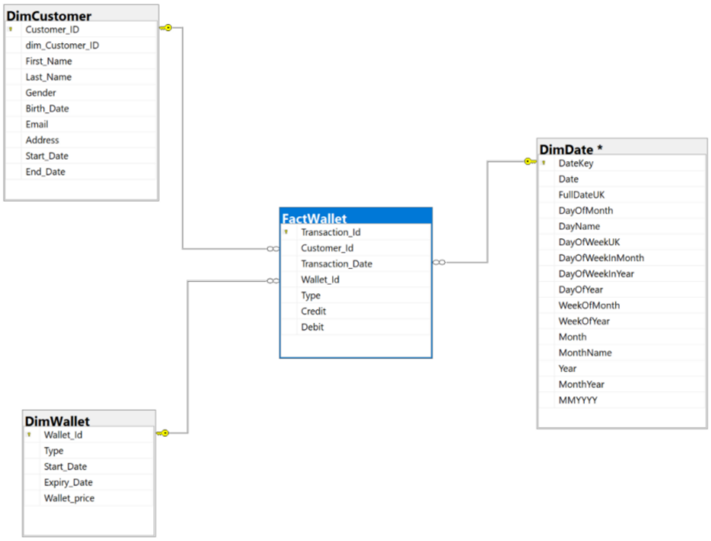
- FactWallet



Columns	Comment
Transaction_Id	Standalone primary key for fact
Customer_ID	Foreign key to DimCustomer
Transaction_Date	Date of transaction and foriengn key to DimDate
Wallet_Id	Foreign key to DimWallet
Type	Type of transaction (Credit, Debit)
Credit	Credit amount
Debit	Debit amount

STAR schema model

Below is the logical diagram of the dimensional model for the eWallet service.



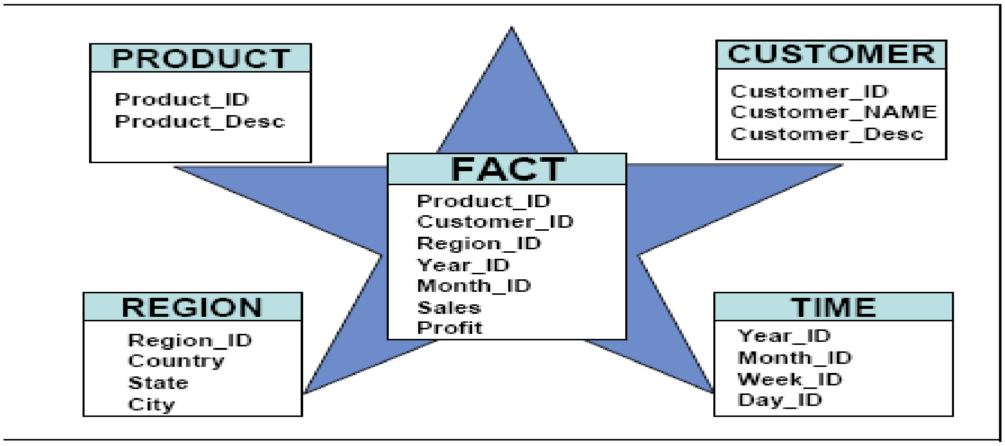
eWallet — Star schema Dimensional Model

Ralph Kimball who is the pioneer in Data warehouse technologies has always shown the importance of Business value in his books. Star schema is preferred over snowflake schema because of more analytical capabilities.

References: Ralph Kimball, Margy Ross, The Data Warehouse Toolkit, 2nd Edition, The complete guide to dimensional modeling

Star Schema

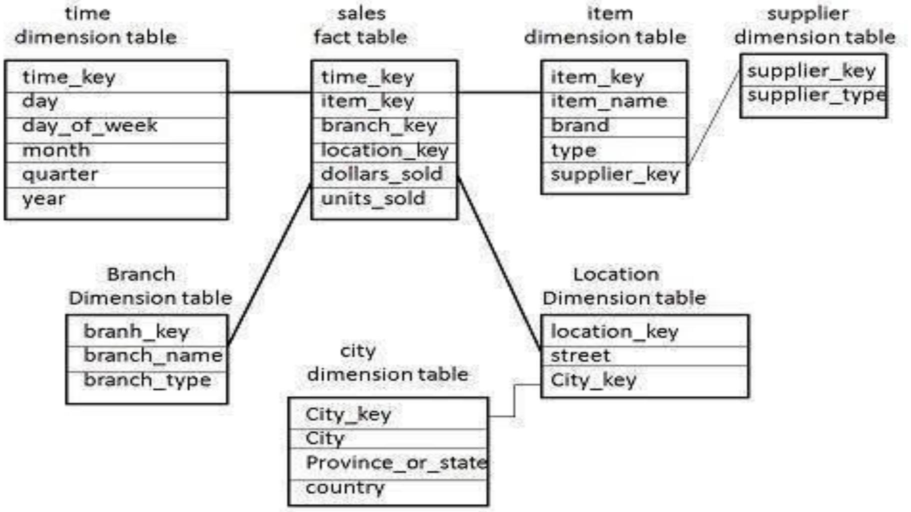
- Fact table is in middle and dimension tables are arranged around the fact table



Snowflake Schema

Normalization and expansion of the dimension tables in a star schema result in the implementation of a snowflake design.

Snowflaking in the dimensional model can impact understandability of the dimensional model and result in a decrease in performance because more tables will need to be joined to satisfy queries



Result:

Conclusion:

We have studied different schemas of data warehouse, and using the methodology of engineering legacy database, a new data warehouse was built. The normalization was applied wherever required on star schema and snowflake schema was designed.

Industrial Application: Applications of OLAP include business reporting for sales, marketing, management reporting, business process management (BPM), budgeting and similar areas, with new applications coming up, such as agriculture The term OLAP was created as a slight modification of the traditional database term online transaction processing.

Questionnaires:

1. Define data warehouse?

2. What is multi-dimensional data?

3. Compare star and snowflake schema?

4. Is a star schema normalized or denormalized?

5. Why do we use snowflake schema

6. Explain Fact table in data warehousing with example.

7. A MDP Stand for________

8. A data warehouse is which of the following?

A. Can be updated by end users. B.Contains numerous naming conventions and formats.

C. Organized around important subject areas. D Contains only current d

Answer:

9. A snowflake schema is which of the following types of tables?

A. Fact B. Dimension C. Helper D. All of the above

Answer:

10. A star schema has One-to-many type of relationship between a dimension and fact table(True or False)?

Answer:

Experiment No. 2

Dimension table and Fact table

Aim: Implementation of all dimension table and fact table based on experiment 1 case study

Objectives: From this experiment, the student will be able to

- What is Fact Table
- What is Dimension Table

Outcomes: The learner will be able to

- Assess the strength and weaknesses of algorithms
- Identify, formulate and solve engineering problems
- Analyse the local and global impact of data mining on individuals, organizations and society

Software Required:

Theory:

Introduction to Fact Table and Dimension Table

Fact tables comprises of the facts of the system as its data content, and Dimension tables comprises of all the properties or objects of the fact tables that can help to connect fact tables to the respective dimension tables. The data in both the tables can be in normal text format, while fact tables can have numbers along with the texts. In the process of creating the database, dimension tables are created before fact tables due to their own properties.

Fact table

It is a table that has values of the attributes of the dimension table. It contains quantitative information in a denormalized form. It basically contains the data that needs to be analyzed. Fact tables mostly have two columns, one for foreign keys that helps to join them with a dimension table and others that contains the value or data that need to be analyzed. It mostly contains numeric data. It grows vertically and it contains more records and fewer attributes.

Characteristics of Fact Table

Keys: Fact table consists of a key that is the combination or concatenation of all primary keys of various dimension tables associated with that fact table. Such key is called a concatenated key which uniquely identifies the row of the fact table.

Fact Table Grain: Grain of the table means the level of the detail or the deepness of the information that is stored in the fact table. The level must be the highest for designing an efficient fact table.

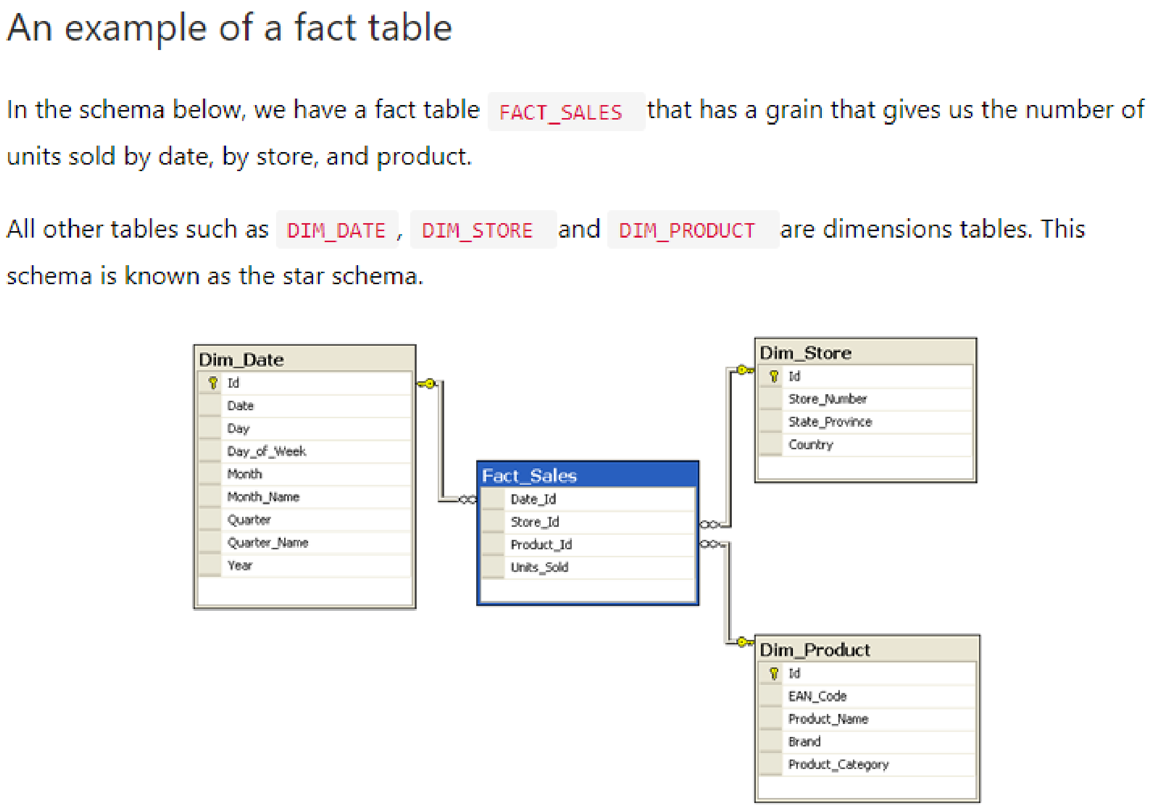
Additive Measures: Attributes in the fact table can be fully additive, semi-additive or non-additive. Fully additive or additive measures are those that are added to all dimensions. In semi-additive, measures are added to some dimensions and not to all and non-additive measures are those which stores basic unit of measurement of any business process.

Sparse Data: Some records present in the fact table contain attributes with null values or measures i.e. these records do not give or provide any information.

Degenerated Dimensions: The dimensions or attributes present in the fact table which cannot be added or which are not additive are called a degenerated dimension.

Outrigger Dimensions: The dimensions that have reference to any other dimension table are called as outrigger dimensions.

Shrunken Rollup Dimensions: The dimensions which are the subdivision of columns and rows of the base dimension are called Shrunken Rollup dimensions.



An example of a fact table

In the schema below, we have a fact table `FACT_SALES` that has a grain that gives us the number of units sold by date, by store, and product.

All other tables such as `DIM_DATE`, `DIM_STORE` and `DIM_PRODUCT` are dimensions tables. This schema is known as the star schema.

Dimension Table

A dimension table contains the dimensions along which the values of the attributes are taken in the fact table. Dimension tables are small in size, contains only several thousand rows but the size can be increased occasionally. These tables are associated with a fact table through foreign keys. These tables are de-normalized. The dimension table contains hierarchical relationships and grows horizontally.

Characteristics of Dimension Table

Keys: Every dimension table needs to have a primary key that helps to uniquely identify each record of the dimension table.

Attributes: Dimension table contains many attributes and therefore the dimension table appears to grow horizontally.

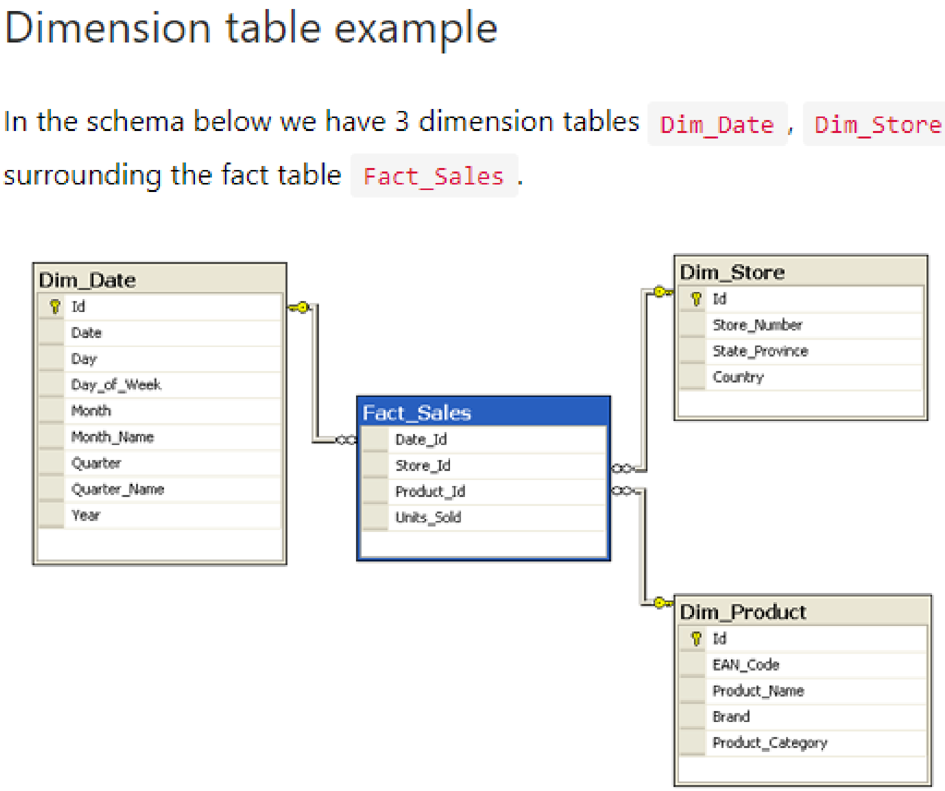
Attribute Values: The values in the dimension table are mostly in textual format and not in numeric format.

Relation Between Attributes: Attributes present in the dimension table are generally not directly related to each other but still are a part of the same dimension table.

Normalization: Dimension table is not normalized because normalization splits the data and creates additional tables which decrease the efficiency of the query execution as it must pass through these additional tables when it wants to recover measurements from the fact table for any corresponding attribute in the dimension table.

Drilling Down, Rolling Up: Attributes present in dimension table permits to derive details through traversing from the higher level to lower level or it also allows rolling up from lower level to the higher level of the attributes.

Records: Dimension table has less number of records and more number of attributes.



Dimension table example

In the schema below we have 3 dimension tables `Dim_Date` , `Dim_Store` and `Dim_Product` surrounding the fact table `Fact_Sales` .

Dim_Date
Id
Date
Day
Day_of_Week
Month
Month_Name
Quarter
Quarter_Name
Year

Fact_Sales
Date_Id
Store_Id
Product_Id
Units_Sold

Dim_Store
Id
Store_Number
State_Province
Country

Dim_Product
Id
EAN_Code
Product_Name
Brand
Product_Category

Fact Table vs Dimension Table Comparison Table

Characteristics	Fact Table	Dimension Table
Basic Definition	It contains measurements, facts or metrics of the attributes.	It is the companion table that contains attributes using which fact table deduce the facts.
Design	It is defined by data grain.	It is descriptive, complete and wordy.
Task	It contains measures and is used for analysis and decision making.	It contains information about a business and its process.
Type of Data	It contains data in both numeric as well as textual format.	It contains data in only textual format.
Key	It has a primary key for each dimension which is acts as a foreign key in the dimension table.	It has a foreign key associated with the primary key of the fact table.
Storage	It stores the filter domain and reports labels in dimension tables.	It stores the detailed atomic data into dimensional structures.
Hierarchy	It does not have a hierarchy.	It contains a hierarchy.
Attributes	It has less attributes	More attributes
Records	More records	Less records.
Table Growth	The table grows vertically.	The table grows horizontally.
Creation Time	A fact table is created after dimension tables are created.	The dimension table needs to be created first.

Schema Structur	There is less number of fact tables in a schema.	There is a number of dimension tables in a schema.

Results:

Conclusion:

Industrial Application:

- Application of Decision Tree Algorithm for Data Mining in Healthcare Operations
- Business Management
- In the past decades, many organizations had created their own databases to enhance their customer services. Decision trees are a possible way to extract useful information from databases and they have already been employed in many applications in the domain of business and management.
- Customer Relationship Management
- A frequently used approach to manage customers' relationships is to investigate how individuals access online services. Such an investigation is mainly performed by collecting and analyzing individuals' usage data and then providing recommendations based on the extracted information. apply decision trees to investigate the relationships between the customers' needs and preferences and the success of online shopping.
- Fraudulent Statement Detection
- Another widely used business application is the detection of Fraudulent Financial Statements (FFS). Such an application is particularly important because it is difficult to discover all hidden information due to the necessity of making a huge number of assumptions and predefining the relationships among the large number of variables in a financial statement.

Questionnaires:

1. A _________ is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.

a)Decision tree b) Graphs c) Trees d) Neural Networks

2.Decision Tree is a display of an algorithm. True or False?

3.What are various classification algorithms?

4.Define entropy?

5.How does u find information gain?

6. How Decision tree works?

7. Decision Trees can be used for Classification Tasks.

a) True

b) False

8. Choose from the following that are Decision Tree nodes

a) Decision Nodes b) End Nodes c) Chance Nodes d) All of the mentioned

9. Decision Nodes are represented by ____________

a) Disks b) Squares c) Circles d) Triangles

10. Chance Nodes are represented by_______

Experiment No. 3

OLAP Operations

Aim: Implementation of OLAP operations: Slice, Dice, Rollup, Drilldown and Pivot based on experiment 1 case study

Objectives: From this experiment, the student will be able to

- What is OLAP
- What are various OLAP Operations

Outcomes: The learner will be able to

- Assess the strength and weaknesses of algorithms
- Identify, formulate and solve engineering problems
- Analyse the local and global impact of data mining on individuals, organizations and society

Software Required:

Theory:

OLAP operations:

In computing, online analytical processing, or OLAP is an approach to answering multi-dimensional analytical (MDA) queries swiftly OLAP is part of the broader category of business intelligence which also encompasses relational database, report writing and data mining. Typical applications of OLAP include business reporting for sales, marketing, management reporting, business process management (BPM), budgeting and similar areas, with new applications coming up, such as agriculture The term OLAP was created as a slight modification of the traditional database term online transaction processing.

Online Analytical Processing Server (OLAP) is based on the multidimensional data model. It allows managers, and analysts to get an insight of the information through fast, consistent, and interactive access to information. This cover the types of OLAP, operations on OLAP, difference between OLAP, and statistical databases and OLTP.

Types of OLAP Servers

We have four types of OLAP servers −

1. Relational OLAP (ROLAP)
2. Multidimensional OLAP (MOLAP)
3. Hybrid OLAP (HOLAP)
4. Specialized SQL Servers

Relational OLAP

ROLAP servers are placed between relational back-end server and client front-end tools. To store and manage warehouse data, ROLAP uses relational or extended-relational DBMS.

ROLAP includes the following −

- Implementation of aggregation navigation logic.
- Optimization for each DBMS back end.
- Additional tools and services.

Since OLAP servers are based on multidimensional view of data, we will discuss OLAP operations in multidimensional data.

Here is the list of OLAP operations −

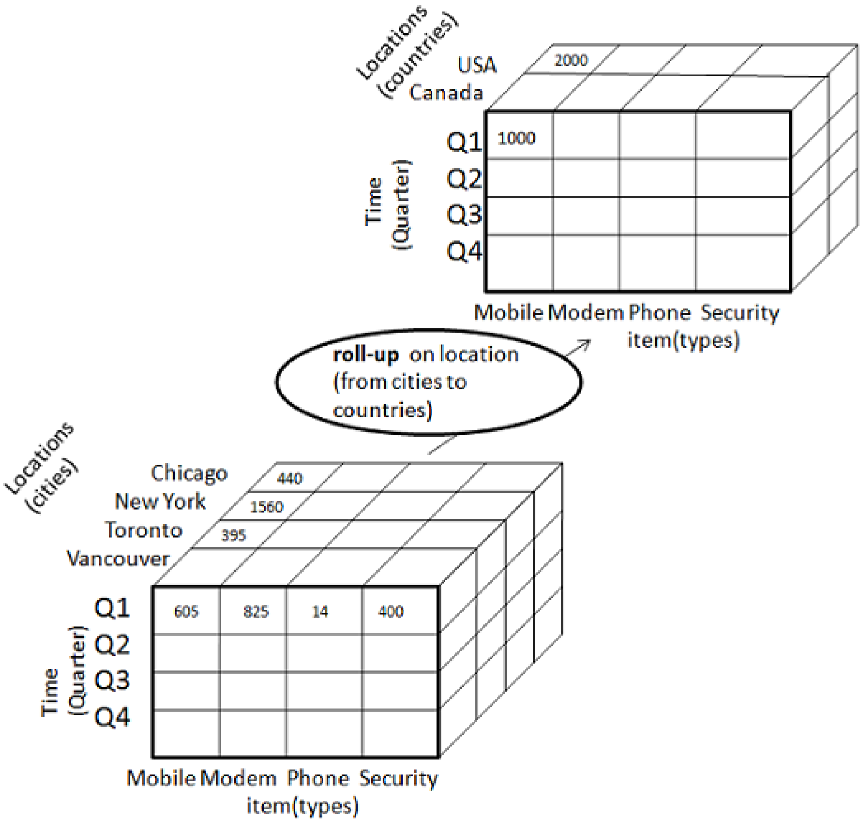
- Roll-up
- Drill-down
- Slice and dice
- Pivot (rotate)

Roll-up

Roll-up performs aggregation on a data cube in any of the following ways −

- By climbing up a concept hierarchy for a dimension
- By dimension reduction

The following diagram illustrates how roll-up works.



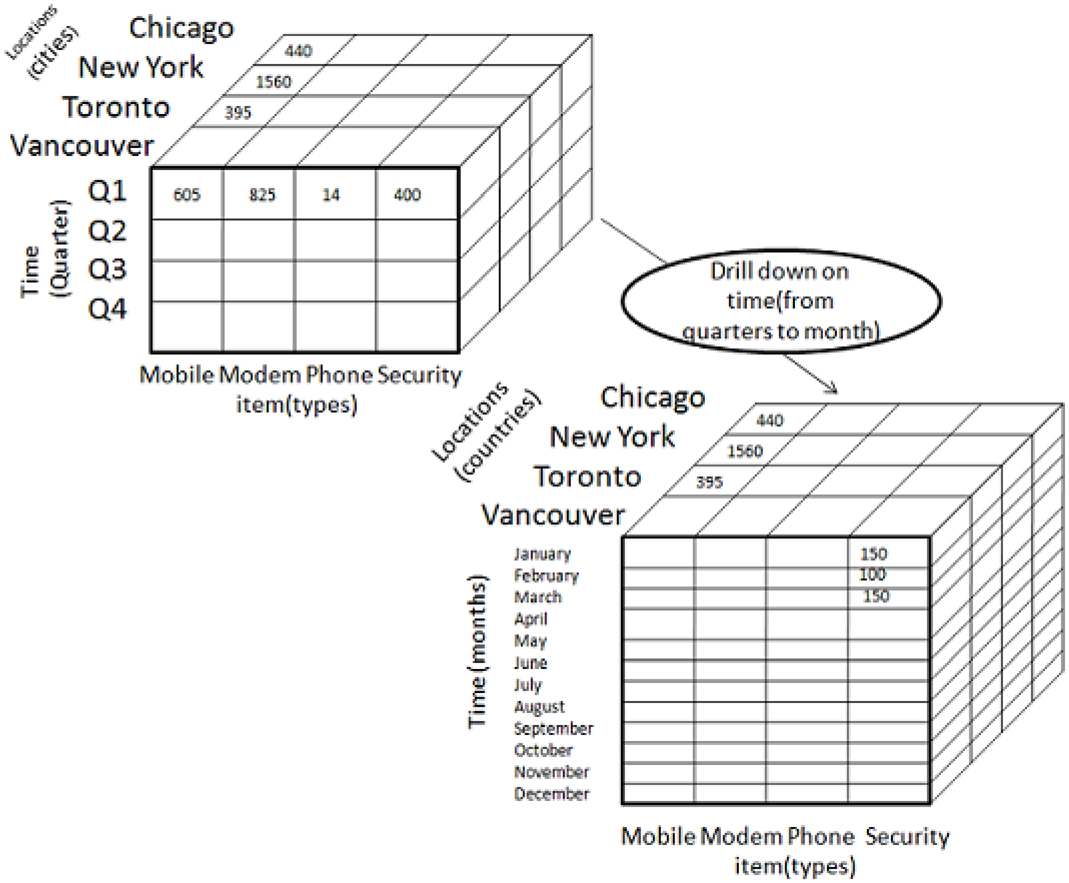
- Roll-up is performed by climbing up a concept hierarchy for the dimension location.
- Initially the concept hierarchy was "street < city < province < country".
- On rolling up, the data is aggregated by ascending the location hierarchy from the level of city to the level of country.
- The data is grouped into cities rather than countries.
- When roll-up is performed, one or more dimensions from the data cube are removed.

Drill-down

Drill-down is the reverse operation of roll-up. It is performed by either of the following ways −

- By stepping down a concept hierarchy for a dimension
- By introducing a new dimension.

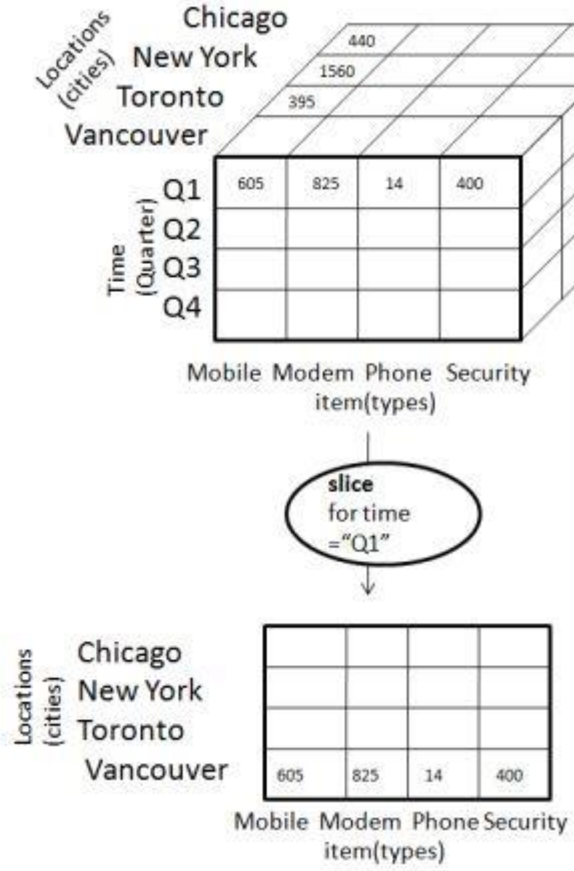
The following diagram illustrates how drill-down works −



- Drill-down is performed by stepping down a concept hierarchy for the dimension time.
- Initially the concept hierarchy was "day < month < quarter < year."
- On drilling down, the time dimension is descended from the level of quarter to the level of month.
- When drill-down is performed, one or more dimensions from the data cube are added.
- It navigates the data from less detailed data to highly detailed data.

Slice

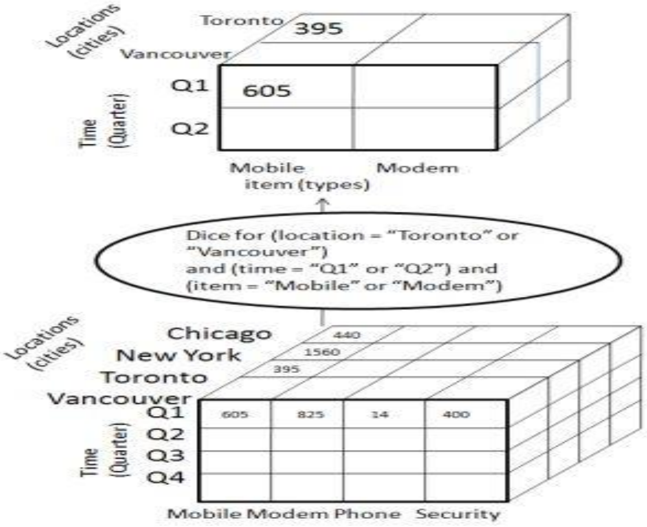
The slice operation selects one particular dimension from a given cube and provides a new sub-cube. Consider the following diagram that shows how slice works.



- Here Slice is performed for the dimension "time" using the criterion time = "Q1".
- It will form a new sub-cube by selecting one or more dimensions.

Dice

Dice selects two or more dimensions from a given cube and provides a new sub-cube. Consider the following diagram that shows the dice operation.

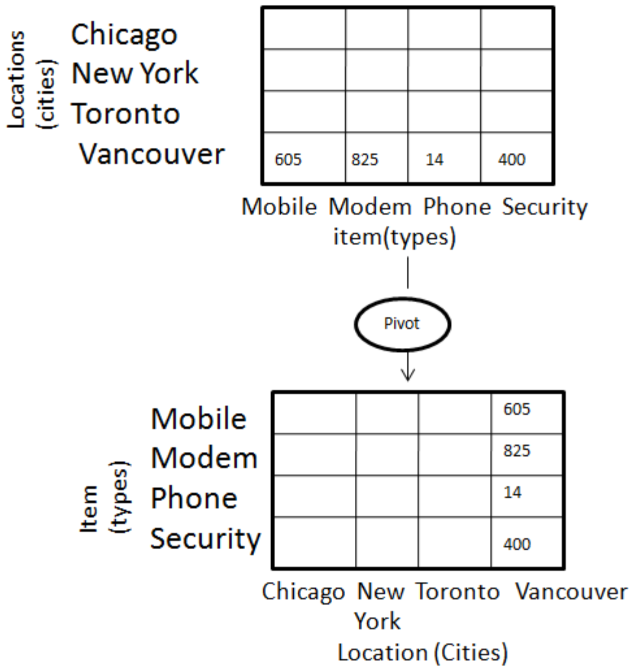


The dice operation on the cube based on the following selection criteria involves three dimensions.

- (location = "Toronto" or "Vancouver")
- (time = "Q1" or "Q2")
- (item =" Mobile" or "Modem")

Pivot

The pivot operation is also known as rotation. It rotates the data axes in view in order to provide an alternative presentation of data. Consider the following diagram that shows the pivot operation.



Results:

Conclusion:

Industrial Application:

Experiment No. 4

File Formats

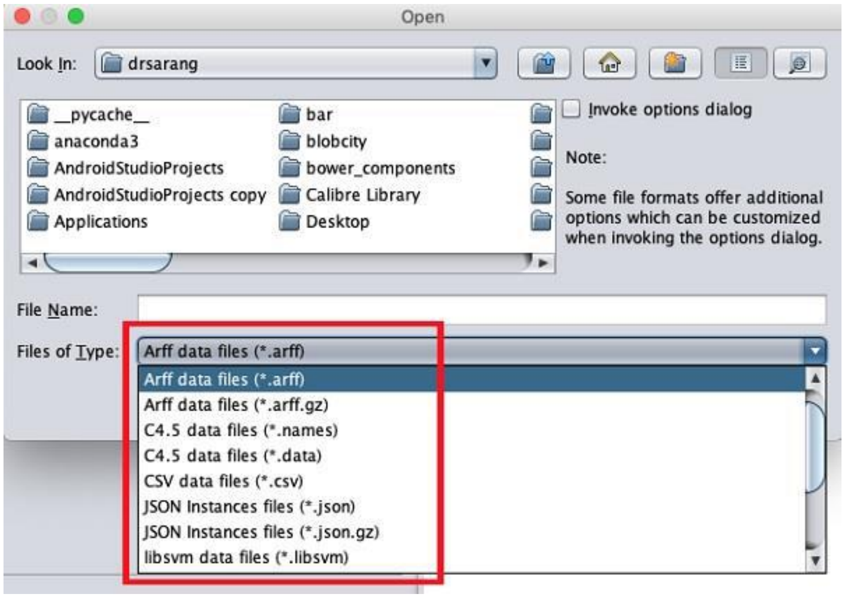
Aim: To study the file formats for the data mining.

Introduction:

WEKA supports a large number of file formats for the data. The complete list of file formats are given here:

1. arff
2. arff.gz
3. bsi
4. csv
5. dat
6. data
7. json
8. json.gz
9. libsvm
10. m
11. names
12. xrff
13. xrff.gz

The types of files that it supports are listed in the drop-down list box at the bottom of the screen.



Open

Look In: drsarang

Note:

Some file formats offer additional options which can be customized when invoking the options dialog.

File Name:

Files of Type: Arff data files (*.arff)

As you would notice it supports several formats including CSV and JSON.

The default file type is Arff.

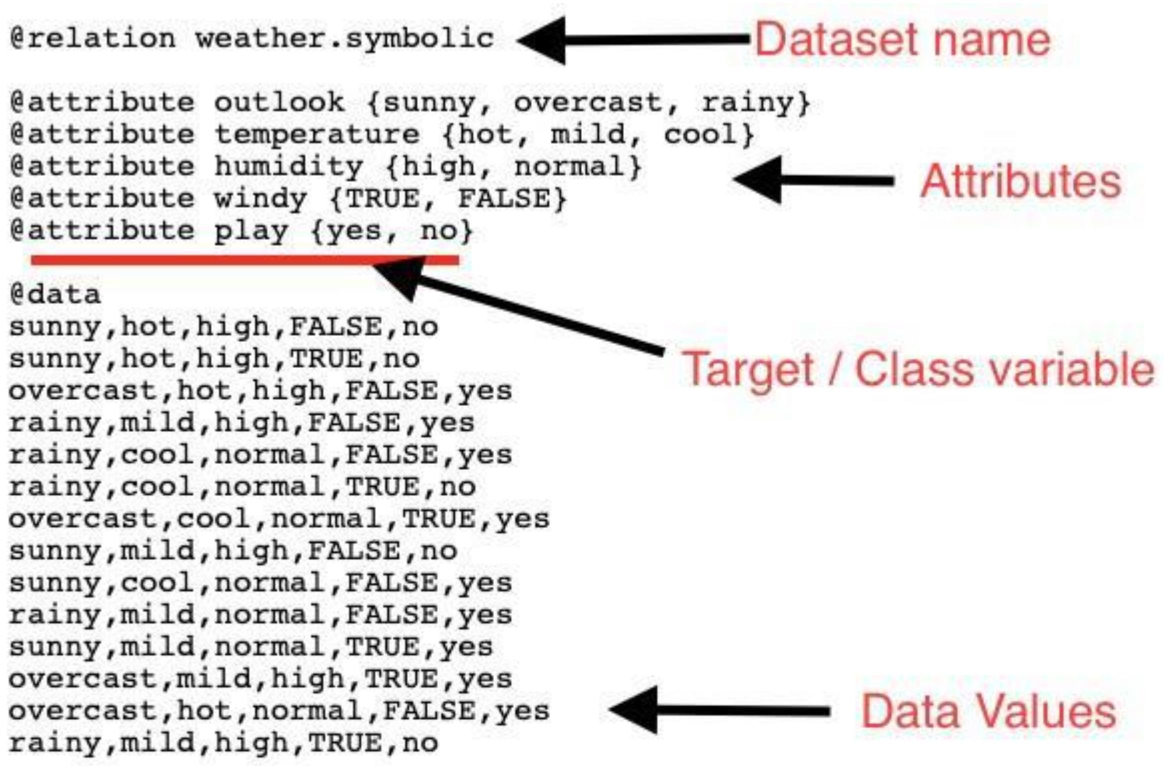
Arff Format

An Arff file contains two sections - header and data.

The header describes the attribute types.

The data section contains a comma separated list of data.

As an example for Arff format, the Weather data file loaded from the WEKA sample databases is shown below:



```
@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```

From the screenshot, you can infer the following points −

The @relation tag defines the name of the database.

The @attribute tag defines the attributes.

The @data tag starts the list of data rows each containing the comma separated fields.

The attributes can take nominal values as in the case of outlook shown here −

@attribute outlook (sunny, overcast, rainy)

The attributes can take real values as in this case −

@attribute temperature real

You can also set a Target or a Class variable called play as shown here −

@attribute play (yes, no)

The Target assumes two nominal values yes or no.

Result:

Thus the different file formats for the data mining was studied.

Experiment No. 5

WEKA Tool

Aim: Create an Employee Table with the help of Data Mining Tool WEKA.

Description:

We need to create an Employee Table with training data set which includes attributes like name, id, salary,

experience, gender, phone number.

Procedure:

Steps:

1) Open Start □ Programs □ Accessories □ Notepad

2) Type the following training data set with the help of Notepad for Employee Table.

```
@relation employee
@attribute name {x,y,z,a,b}
@attribute id numeric
@attribute salary {low,medium,high}
@attribute exp numeric
@attribute gender {male,female}
@attribute phone numeric
@data
x,101,low,2,male,250311
y,102,high,3,female,251665
z,103,medium,1,male,240238
a,104,low,5,female,200200
b,105,high,2,male,240240
```

3) After that the file is saved with **.arff** file format.

4) Minimize the arff file and then open Start □ Programs □ weka-3-4.

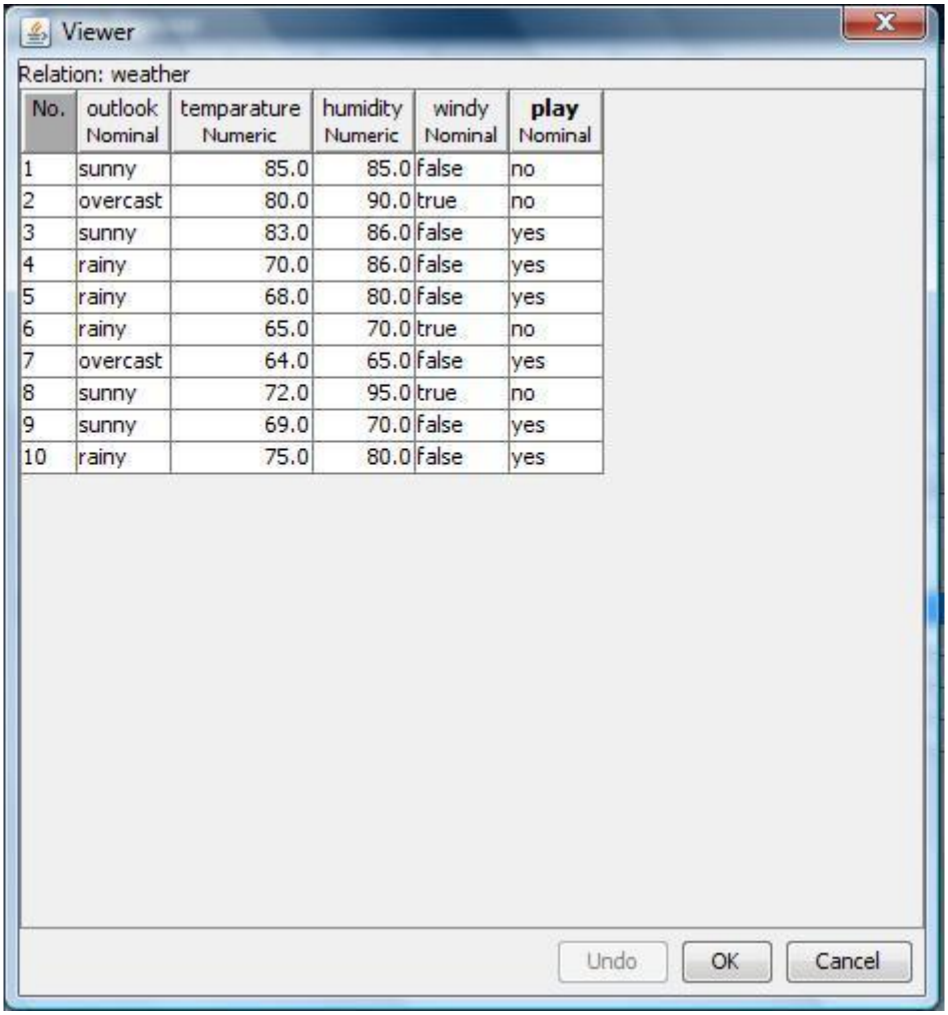
5) Click on **weka-3-4**, then Weka dialog box is displayed on the screen.

6) In that dialog box there are four modes, click on **explorer**.

7) Explorer shows many options. In that click on **‘open file’** and select the arff file

8) Click on **edit button** which shows employee table on weka.3

Training Data Set Weather Table



Relation: weather

No.	outlook Nominal	temparature Numeric	humidity Numeric	windy Nominal	play Nominal
1	sunny	85.0	85.0	false	no
2	overcast	80.0	90.0	true	no
3	sunny	83.0	86.0	false	yes
4	rainy	70.0	86.0	false	yes
5	rainy	68.0	80.0	false	yes
6	rainy	65.0	70.0	true	no
7	overcast	64.0	65.0	false	yes
8	sunny	72.0	95.0	true	no
9	sunny	69.0	70.0	false	yes
10	rainy	75.0	80.0	false	yes

Result:

This program has been successfully executed

Experiment No. 6

WEKA Tool

Aim: Apply Pre-Processing techniques to the training data set of Weather Table

Description:

Real world databases are highly influenced to noise, missing and inconsistency due to their queue size so the

data can be pre-processed to improve the quality of data and missing results and it also improves the efficiency.

There are 3 pre-processing techniques they are:

1) Add

2) Remove

3) Normalization

Creation of Weather Table:

Procedure:

1) Open Start □ Programs □ Accessories □ Notepad

2) Type the following training data set with the help of Notepad for Weather Table.

```
@relation weather

@attribute outlook {sunny,rainy,overcast}

@attribute temparature numeric

@attribute humidity numeric

@attribute windy {true,false}

@attribute play {yes,no}

@data

sunny,85.0,85.0,false,no

overcast,80.0,90.0,true,no

sunny,83.0,86.0,false,yes
```

rainy,70.0,86.0,false,yes

rainy,68.0,80.0,false,yes

rainy,65.0,70.0,true,no

overcast,64.0,65.0,false,yes

sunny,72.0,95.0,true,no

sunny,69.0,70.0,false,yes

rainy,75.0,80.0,false,yes

3) After that the file is saved with **.arff** file format.

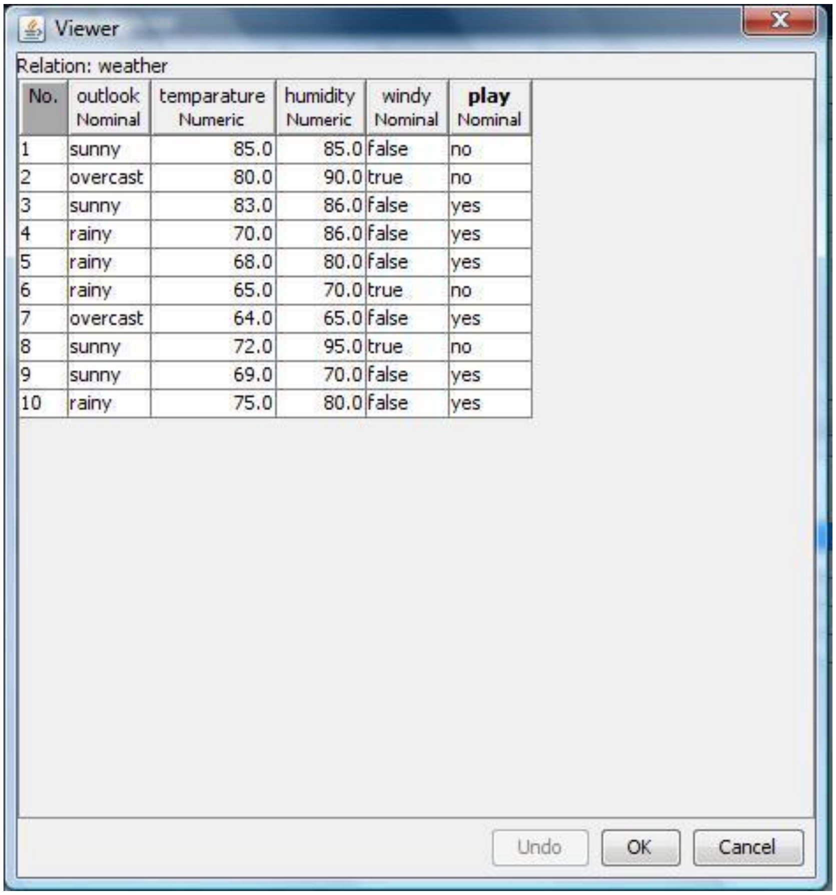
4) Minimize the arff file and then open Start □ Programs □ weka-3-4.

5) Click on **weka-3-4**, then Weka dialog box is displayed on the screen.

6) In that dialog box there are four modes, click on **explorer**.

7) Explorer shows many options. In that click on **'open file'** and select the arff file

8) Click on **edit button** which shows weather table on weka.7



Relation: weather

No.	outlook Nominal	temparature Numeric	humidity Numeric	windy Nominal	**play** Nominal
1	sunny	85.0	85.0	false	no
2	overcast	80.0	90.0	true	no
3	sunny	83.0	86.0	false	yes
4	rainy	70.0	86.0	false	yes
5	rainy	68.0	80.0	false	yes
6	rainy	65.0	70.0	true	no
7	overcast	64.0	65.0	false	yes
8	sunny	72.0	95.0	true	no
9	sunny	69.0	70.0	false	yes
10	rainy	75.0	80.0	false	yes

Add Pre-Processing Technique:

Procedure:

1) Start Programs Weka-3-4 Weka-3-4

2) Click on **explorer.**

3) Click on **open file.**

4) Select **Weather.arff** file and click on open.

5) Click on **Choose button** and select the **Filters option**.

6) In Filters, we have **Supervised** and **Unsupervised data**.

7) Click on **Unsupervised data**.

8) Select the attribute **Add**.

9) A new window is opened.

10) In that we enter attribute index, type, data format, nominal label values for **Climate**.

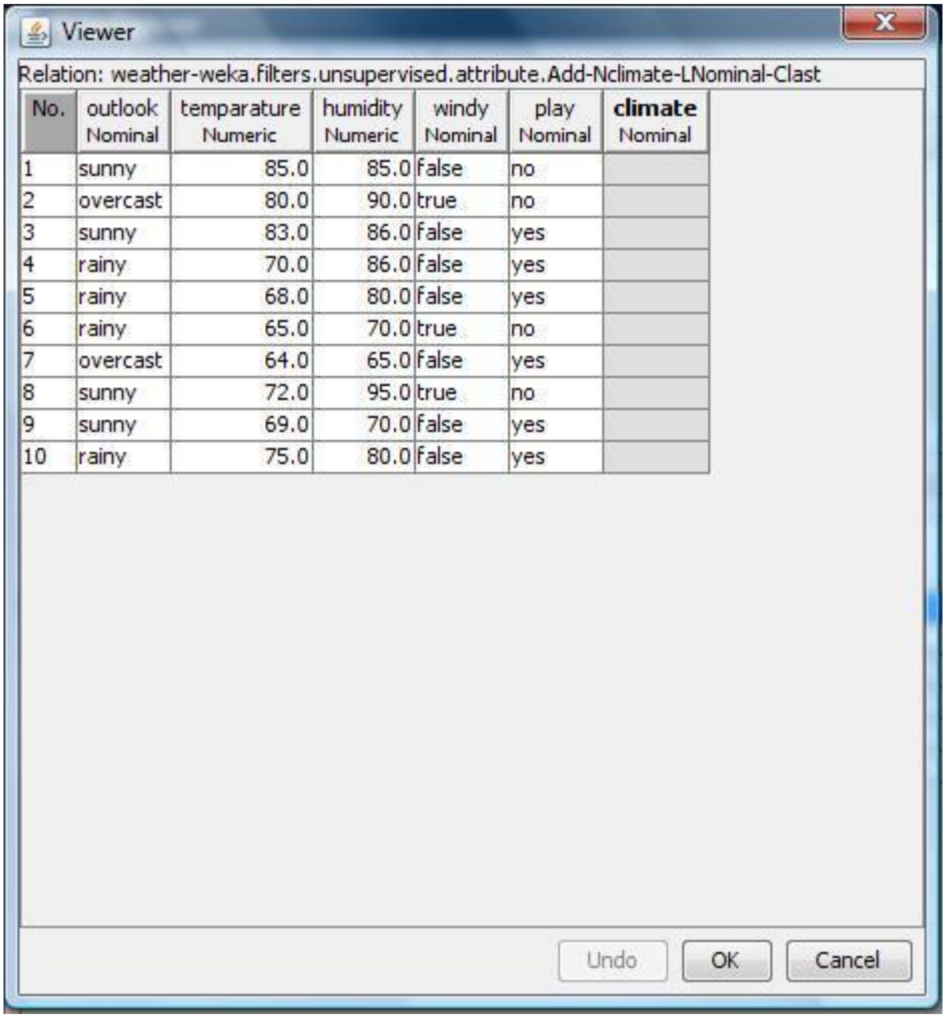
11) Click on **OK**.

12) Press the **Apply button**, then a new attribute is added to the Weather Table.

13) Save the file.

14) Click on the **Edit button**, it shows a new Weather Table on Weka.

Weather Table after adding new attribute CLIMATE:



Relation: weather-weka.filters.unsupervised.attribute.Add-Nclimate-LNominal-Clast

No.	outlook Nominal	temparature Numeric	humidity Numeric	windy Nominal	play Nominal	climate Nominal
1	sunny	85.0	85.0	false	no	
2	overcast	80.0	90.0	true	no	
3	sunny	83.0	86.0	false	yes	
4	rainy	70.0	86.0	false	yes	
5	rainy	68.0	80.0	false	yes	
6	rainy	65.0	70.0	true	no	
7	overcast	64.0	65.0	false	yes	
8	sunny	72.0	95.0	true	no	
9	sunny	69.0	70.0	false	yes	
10	rainy	75.0	80.0	false	yes	

Remove Pre-Processing Technique:

Procedure:

1) Start □ Programs □ Weka-3-4 □ Weka-3-4

2) Click on **explorer.**

3) Click on **open file.**

4) Select **Weather.arff** file and click on open.

5) Click on **Choose button** and select the **Filters option**.

6) In Filters, we have **Supervised** and **Unsupervised data**.

7) Click on **Unsupervised data**.

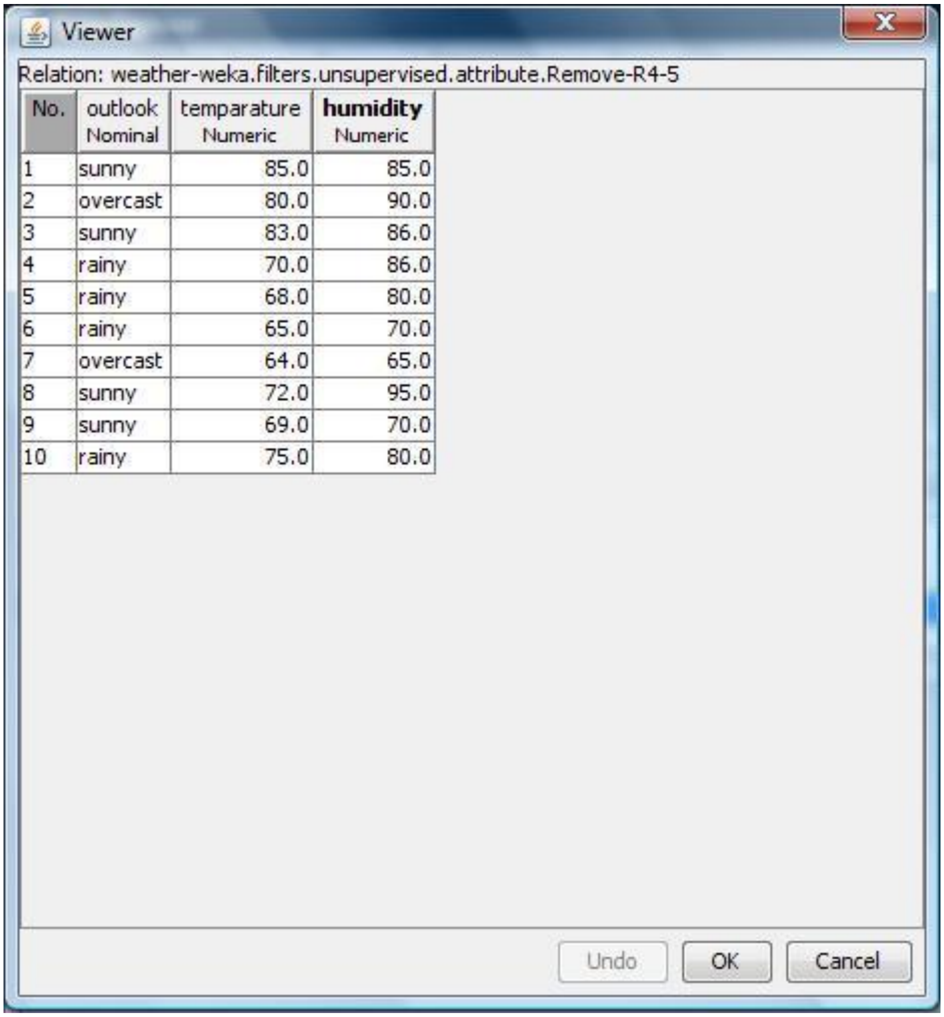
8) Select the attribute **Remove**.

9) Select the attributes **windy, play** to Remove.

10) Click **Remove button** and then **Save**.

11) Click on the **Edit button**, it shows a new Weather Table on Weka.9

Weather Table after removing attributes WINDY, PLAY:



Relation: weather-weka.filters.unsupervised.attribute.Remove-R4-5

No.	outlook Nominal	temparature Numeric	humidity Numeric
1	sunny	85.0	85.0
2	overcast	80.0	90.0
3	sunny	83.0	86.0
4	rainy	70.0	86.0
5	rainy	68.0	80.0
6	rainy	65.0	70.0
7	overcast	64.0	65.0
8	sunny	72.0	95.0
9	sunny	69.0	70.0
10	rainy	75.0	80.0

Normalize Pre-Processing Technique:

Procedure:

1) Start Programs Weka-3-4 Weka-3-4

2) Click on **explorer.**

3) Click on **open file.**

4) Select **Weather.arff** file and click on open.

5) Click on **Choose button** and select the **Filters option**.

6) In Filters, we have **Supervised** and **Unsupervised data**.

7) Click on **Unsupervised data**.

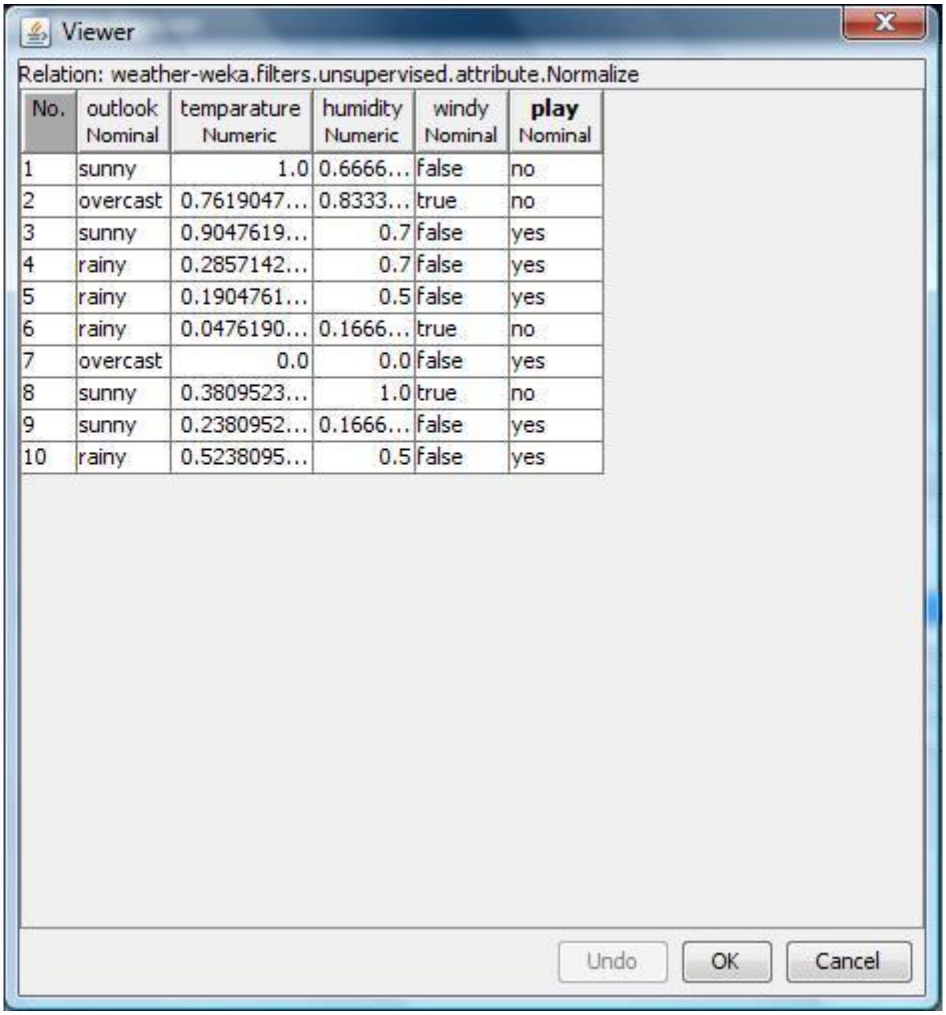
8) Select the attribute **Normalize**.

9) Select the attributes **temparature, humidity** to Normalize.

10) Click on **Apply button** and then **Save**.

11) Click on the **Edit button**, it shows a new Weather Table with normalized values on Weka.

Weather Table after Normalizing TEMPARATURE, HUMIDITY:



Relation: weather-weka.filters.unsupervised.attribute.Normalize

No.	outlook Nominal	temparature Numeric	humidity Numeric	windy Nominal	play Nominal
1	sunny	1.0	0.6666...	false	no
2	overcast	0.7619047...	0.8333...	true	no
3	sunny	0.9047619...	0.7	false	yes
4	rainy	0.2857142...	0.7	false	yes
5	rainy	0.1904761...	0.5	false	yes
6	rainy	0.0476190...	0.1666...	true	no
7	overcast	0.0	0.0	false	yes
8	sunny	0.3809523...	1.0	true	no
9	sunny	0.2380952...	0.1666...	false	yes
10	rainy	0.5238095...	0.5	false	yes

Result:

This program has been successfully executed.

Experiment No. 7

WEKA Tool

Decision tree, Naïve Bayes, Random Forest using WEKA

Aim: To implement like Decision tree, Naïve Bayes, Random Forest using WEKA

Objectives: From this experiment, the student will be able to

- Analyse the data, identify the problem and choose relevant algorithm to apply
- Understand and implement classical algorithms in data mining
- Identify the application of classification algorithm in data mining

Outcomes: The learner will be able to

- Assess the strength and weaknesses of algorithms
- Identify, formulate and solve engineering problems
- Analyse the local and global impact of data mining on individuals, organizations and society

Software Required :WEKA tool

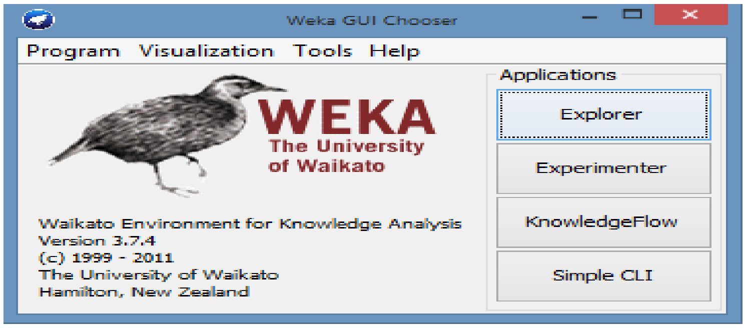
Theory:

Weka is a landmark system in the history of the data mining and machine learning research communities,because it is the only toolkit that has gained such widespread adoption and survived for an extended period of time

The key features responsible for Weka's success are: –

- It provides many different algorithms for data mining and machine learning.
- Is is open source and freely available.
- It is platform-independent.
- It is easily useable by people who are not data mining specialists.
- It provides flexible facilities for scripting experiments – it has kept up-to-date, with new algorithms

WEKA INTERFACE



Waikato Environment for Knowledge Analysis
Version 3.7.4
(c) 1999 - 2011
The University of Waikato
Hamilton, New Zealand

The GUI Chooser consists of four buttons—one for each of the four major Weka applications—and four menus.The buttons can be used to start the following applications:

• Explorer : An environment for exploring data with WEKA .

• Experimenter : An environment for performing experiments and conducting statistical tests between learning schemes.

• KnowledgeFlow : This environment supports essentially the same functions as the Explorer but with a drag-and-drop interface. One advantage is that it supports incremental learning.

• SimpleCLI : Provides a simple command-line interface that allows direct execution of WEKA commands for operating systems that do not provide their own command line interface.

Decision tree learning is a method for assessing the most likely outcome value by taking into account the known values of the stored data instances. This learning method is among the most popular of inductive inference algorithms and has been successfully applied in broad range of tasks such as assessing the credit risk of applicants and improving loyality of regular customers

Classification models are tested by comparing the predicted values to known target values in a set of test data. The historical data for a classification project is typically divided into two data sets: one for building the model; the other for testing the model. Scoring a classification model results in class assignments and probabilities for each case. For example, a model that classifies customers as low, medium, or high value would also predict the probability of each classification for each customer. Classification has many applications in customer segmentation, business modeling, marketing, credit analysis, and biomedical and drug response modeling.

Different Classification Algorithms: Oracle Data Mining provides the following algorithms for classification:

□ Decision Tree - Decision trees automatically generate rules, which are conditional statements that reveal the logic used to build the tree.

□ Naive Bayes - Naive Bayes uses Bayes' Theorem, a formula that calculates a probability by counting the frequency of values and combinations of values in the historical data.

PROCEDURE:

1) Open Weka GUI Chooser.

2) Select EXPLORER present in Applications.

3) Select Preprocess Tab.

4) Go to OPEN file and browse the file that is already stored in the system “bank.csv”.

5) Go to Classify tab.

6) Here the c4.5 algorithm has been chosen which is entitled as j48 in Java and can be selected by clicking the button choose and select tree j48

7) Select Test options “Use training set”

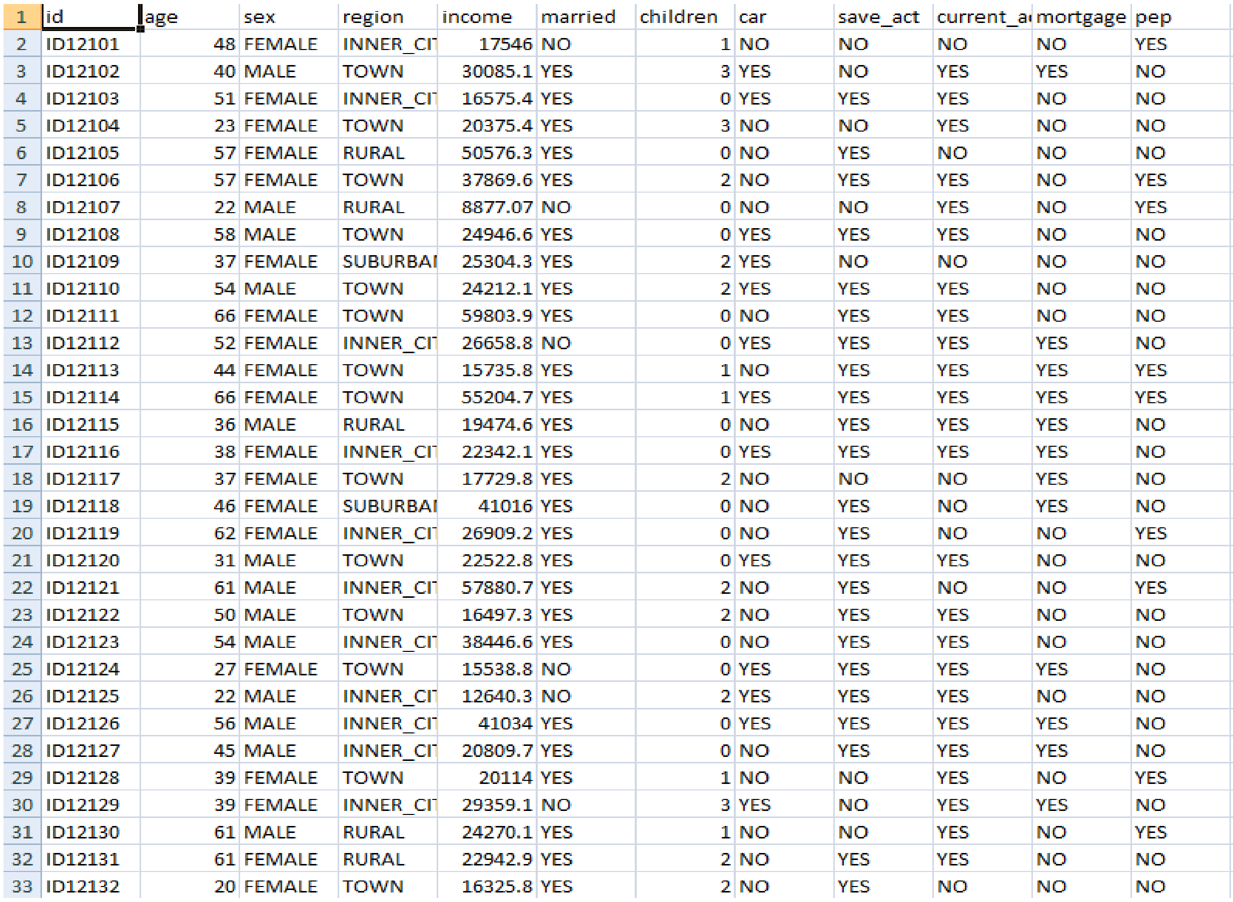
8) if need select attribute.

9) Click Start.

10) Now we can see the output details in the Classifier output.

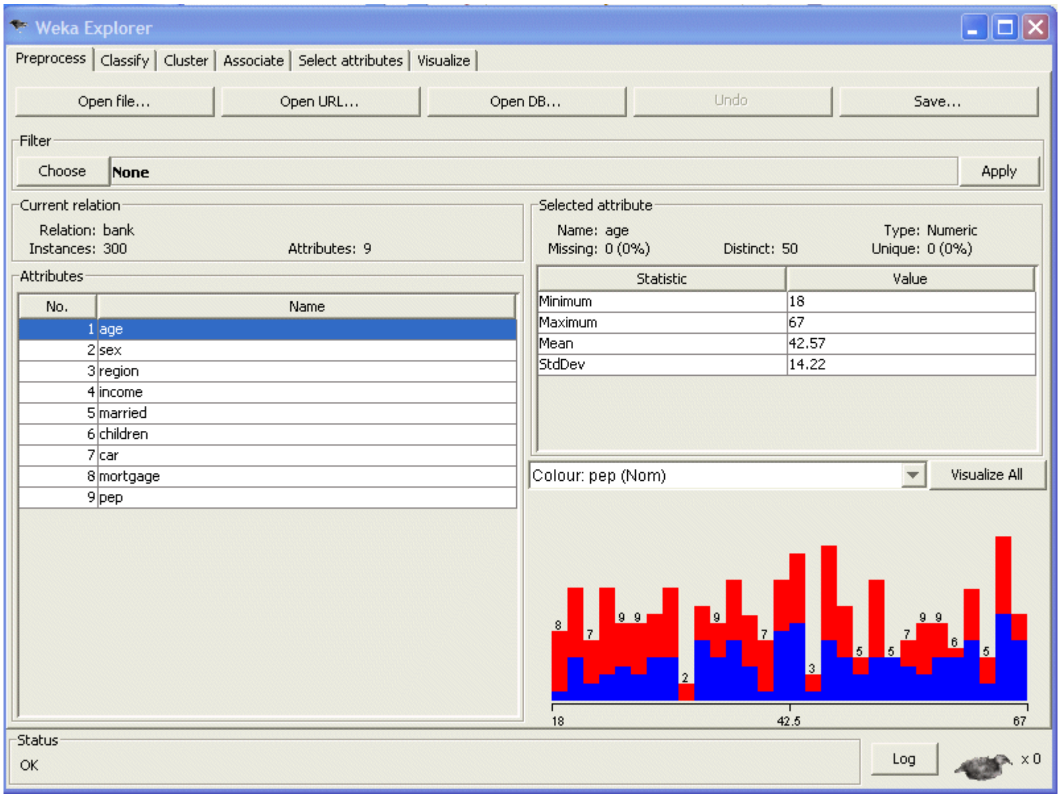
11) Right click on the result list and select “visualize tree”option .

1. Download data set for implementation of ID3 algorithm (.csv or .arff file). Here bank-data.csv data set has taken for decision tree analysis



1	id	age	sex	region	income	married	children	car	save_act	current_a	mortgage	pep
2	ID12101	48	FEMALE	INNER_CI	17546	NO	1	NO	NO	NO	NO	YES
3	ID12102	40	MALE	TOWN	30085.1	YES	3	YES	NO	YES	YES	NO
4	ID12103	51	FEMALE	INNER_CI	16575.4	YES	0	YES	YES	YES	NO	NO
5	ID12104	23	FEMALE	TOWN	20375.4	YES	3	NO	NO	YES	NO	NO
6	ID12105	57	FEMALE	RURAL	50576.3	YES	0	NO	YES	NO	NO	NO
7	ID12106	57	FEMALE	TOWN	37869.6	YES	2	NO	YES	YES	NO	YES
8	ID12107	22	MALE	RURAL	8877.07	NO	0	NO	NO	YES	NO	YES
9	ID12108	58	MALE	TOWN	24946.6	YES	0	YES	YES	YES	NO	NO
10	ID12109	37	FEMALE	SUBURBAI	25304.3	YES	2	YES	NO	NO	NO	NO
11	ID12110	54	MALE	TOWN	24212.1	YES	2	YES	YES	YES	NO	NO
12	ID12111	66	FEMALE	TOWN	59803.9	YES	0	NO	YES	YES	NO	NO
13	ID12112	52	FEMALE	INNER_CI	26658.8	NO	0	YES	YES	YES	YES	NO
14	ID12113	44	FEMALE	TOWN	15735.8	YES	1	NO	YES	YES	YES	YES
15	ID12114	66	FEMALE	TOWN	55204.7	YES	1	YES	YES	YES	YES	YES
16	ID12115	36	MALE	RURAL	19474.6	YES	0	NO	YES	YES	YES	NO
17	ID12116	38	FEMALE	INNER_CI	22342.1	YES	0	YES	YES	YES	YES	NO
18	ID12117	37	FEMALE	TOWN	17729.8	YES	2	NO	NO	NO	YES	NO
19	ID12118	46	FEMALE	SUBURBAI	41016	YES	0	NO	YES	NO	YES	NO
20	ID12119	62	FEMALE	INNER_CI	26909.2	YES	0	NO	YES	NO	NO	YES
21	ID12120	31	MALE	TOWN	22522.8	YES	0	YES	YES	YES	NO	NO
22	ID12121	61	MALE	INNER_CI	57880.7	YES	2	NO	YES	NO	NO	YES
23	ID12122	50	MALE	TOWN	16497.3	YES	2	NO	YES	YES	NO	NO
24	ID12123	54	MALE	INNER_CI	38446.6	YES	0	NO	YES	YES	NO	NO
25	ID12124	27	FEMALE	TOWN	15538.8	NO	0	YES	YES	YES	YES	NO
26	ID12125	22	MALE	INNER_CI	12640.3	NO	2	YES	YES	YES	NO	NO
27	ID12126	56	MALE	INNER_CI	41034	YES	0	YES	YES	YES	YES	NO
28	ID12127	45	MALE	INNER_CI	20809.7	YES	0	NO	YES	YES	YES	NO
29	ID12128	39	FEMALE	TOWN	20114	YES	1	NO	NO	YES	NO	YES
30	ID12129	39	FEMALE	INNER_CI	29359.1	NO	3	YES	NO	YES	YES	NO
31	ID12130	61	MALE	RURAL	24270.1	YES	1	NO	NO	YES	NO	YES
32	ID12131	61	FEMALE	RURAL	22942.9	YES	2	NO	YES	YES	NO	NO
33	ID12132	20	FEMALE	TOWN	16325.8	YES	2	NO	YES	NO	NO	NO

2. Load data in WEKA tool



Filter: None

Current relation

Relation: bank
Instances: 300 Attributes: 9

Attributes

No.	Name
1	age
2	sex
3	region
4	income
5	married
6	children
7	car
8	mortgage
9	pep

Selected attribute

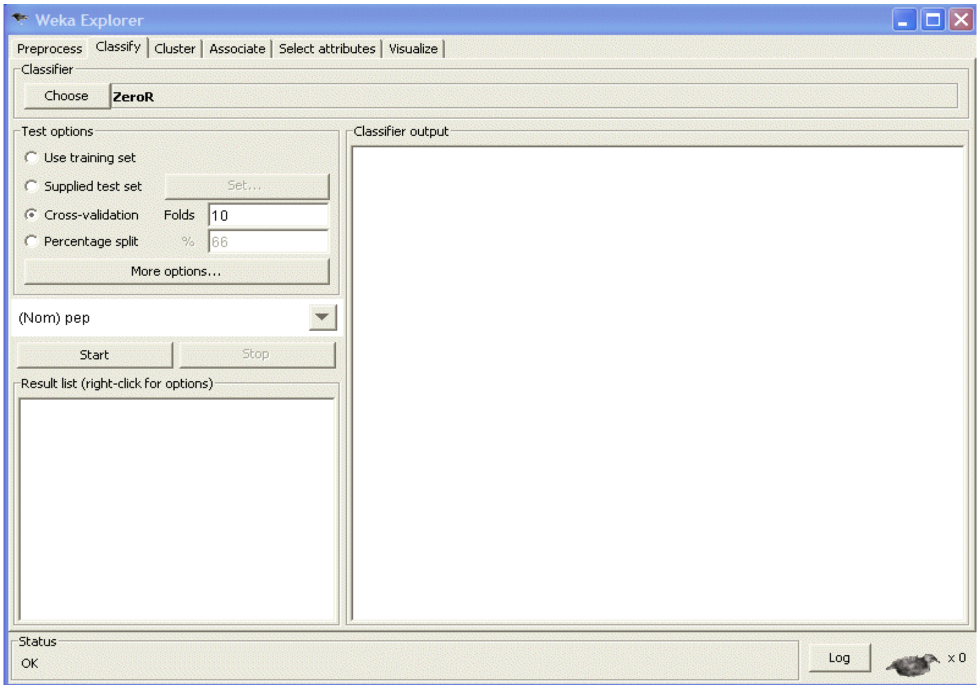
Name: age Type: Numeric
Missing: 0 (0%) Distinct: 50 Unique: 0 (0%)

Statistic	Value
Minimum	18
Maximum	67
Mean	42.57
StdDev	14.22

Colour: pep (Nom)

Status
OK

3. Select the "Classify" tab and click the "Choose" button to select the ID3 classifier



Classifier: ZeroR

Test options

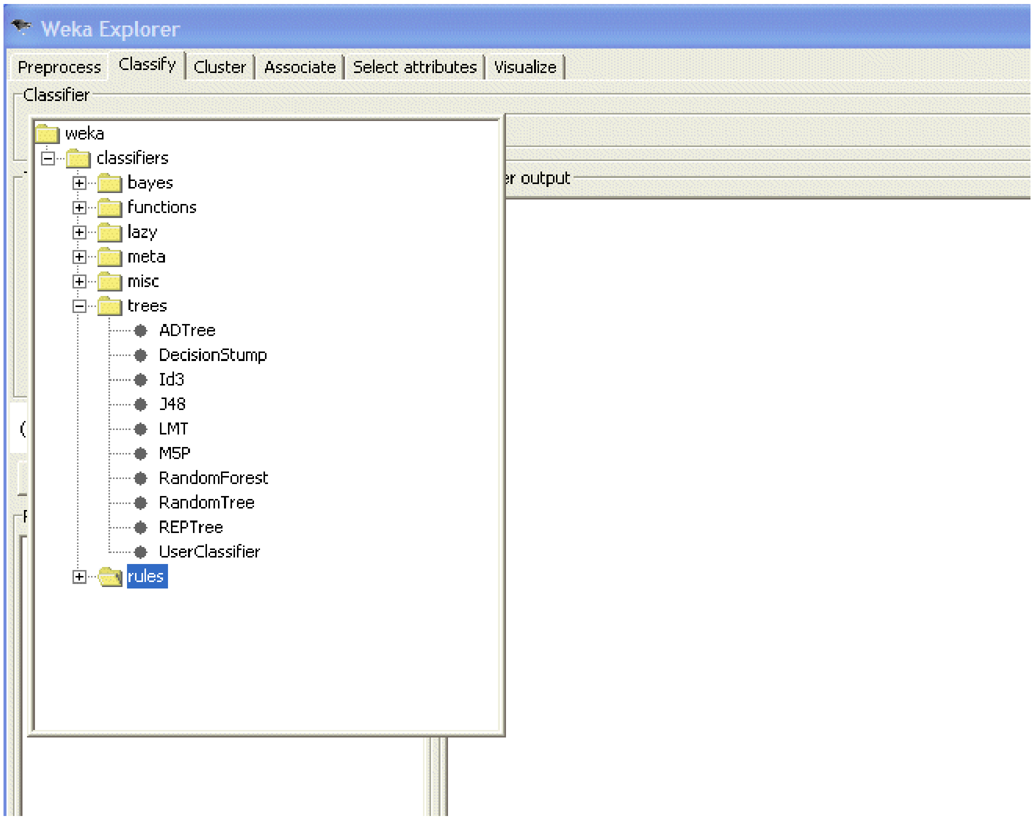
○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) pep

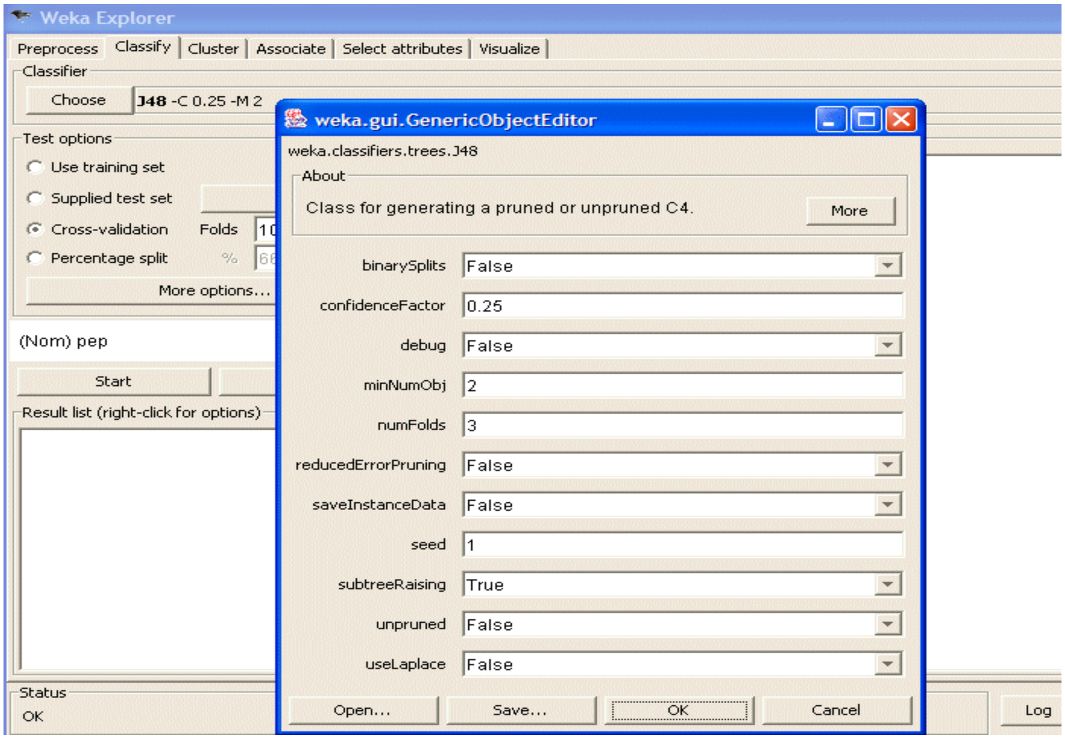
Result list (right-click for options)

Classifier output

Status
OK



1. Specify the various parameters. These can be specified by clicking in the text box to the right of the "Choose" button. In this example we accept the default values. The default version does perform some pruning (using the subtree raising approach), but does not perform error pruning



J48 -C 0.25 -M 2

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66
More options...

(Nom) pep

weka.classifiers.trees.J48

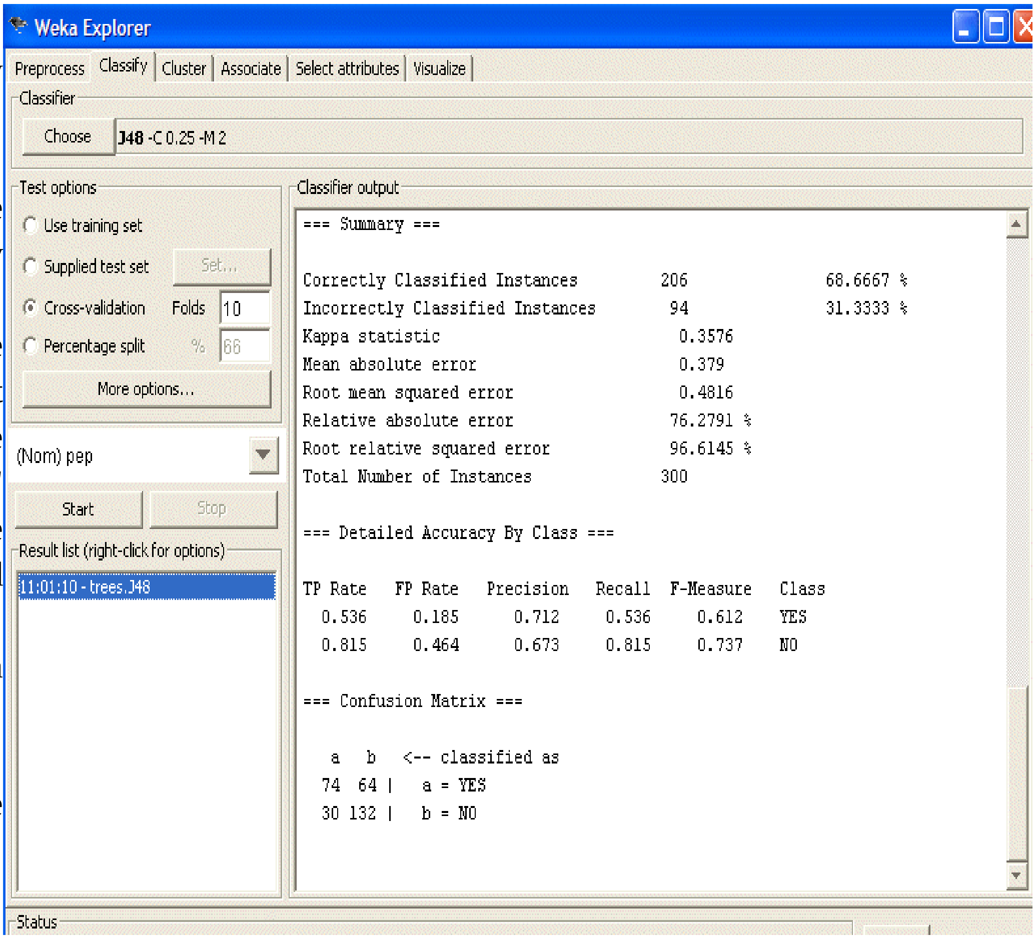
Class for generating a pruned or unpruned C4.

binarySplits	False
confidenceFactor	0.25
debug	False
minNumObj	2
numFolds	3
reducedErrorPruning	False
saveInstanceData	False
seed	1
subtreeRaising	True
unpruned	False
useLaplace	False

2. Under the "Test options" in the main panel we select 10-fold cross-validation as our evaluation approach. Since we do not have separate evaluation data set, this is

necessary to get a reasonable idea of accuracy of the generated model. We now click "Start" to generate the model.

3. We can view this information in a separate window by right clicking the last result set (inside the "Result list" panel on the left) and selecting "View in separate window" from the pop-up menu.



J48 -C 0.25 -M 2

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

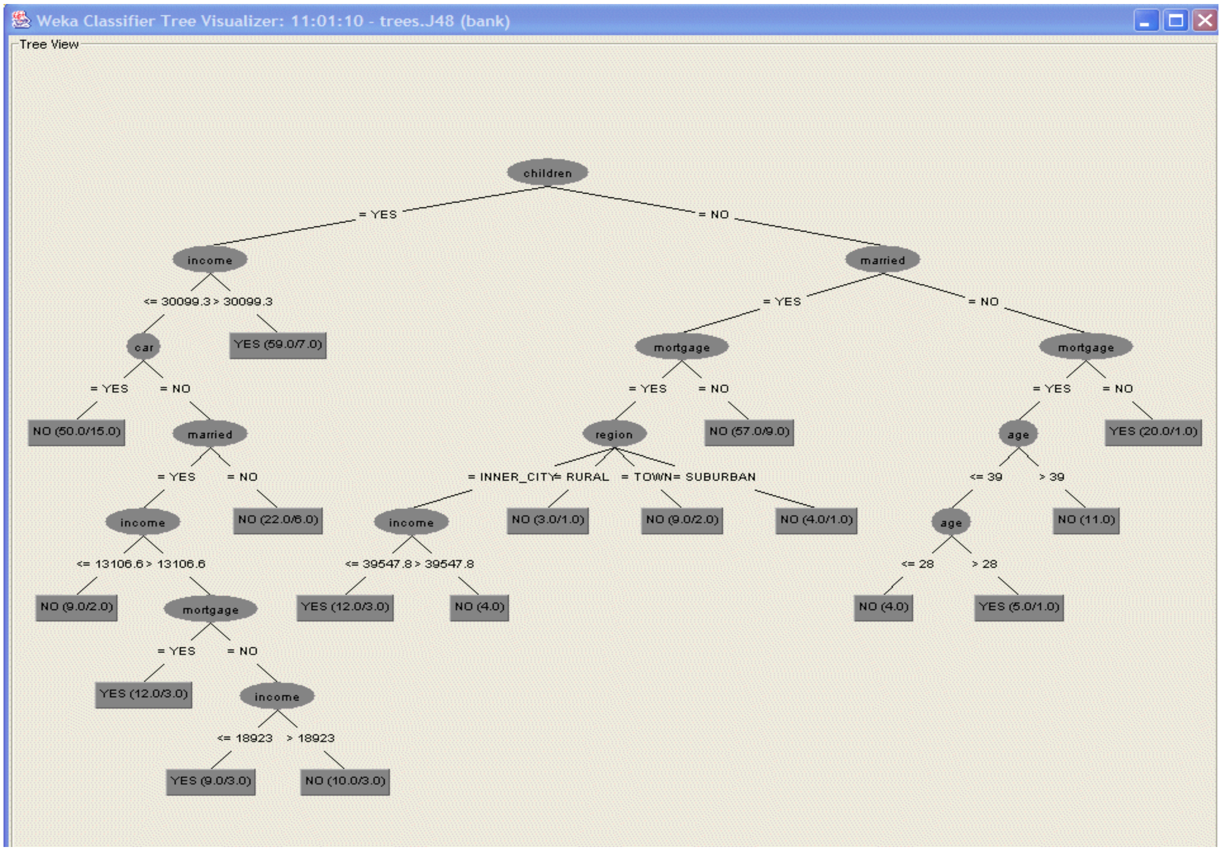
(Nom) pep

Result list (right-click for options)

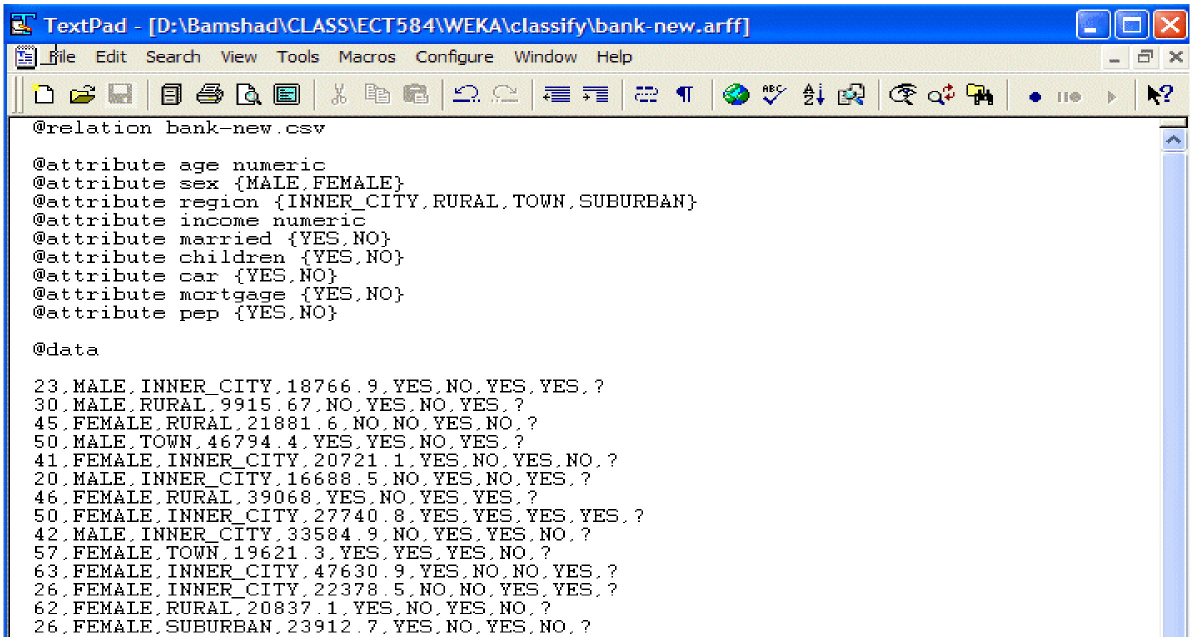
11:01:10 - trees.J48

```
=== Summary ===

Correctly Classified Instances         206               68.6667 %
Incorrectly Classified Instances        94               31.3333 %
Kappa statistic                          0.3576
Mean absolute error                      0.379
Root mean squared error                  0.4816
Relative absolute error                 76.2791 %
Root relative squared error             96.6145 %
Total Number of Instances              300

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.536     0.185      0.712     0.536     0.612    YES
  0.815     0.464      0.673     0.815     0.737    NO

=== Confusion Matrix ===

   a   b   <-- classified as
  74  64 |   a = YES
  30 132 |   b = NO
```

4. WEKA also provides view a graphical rendition of the classification tree. This can be done by right clicking the last result set (as before) and selecting "Visualize tree" from the pop-up menu.

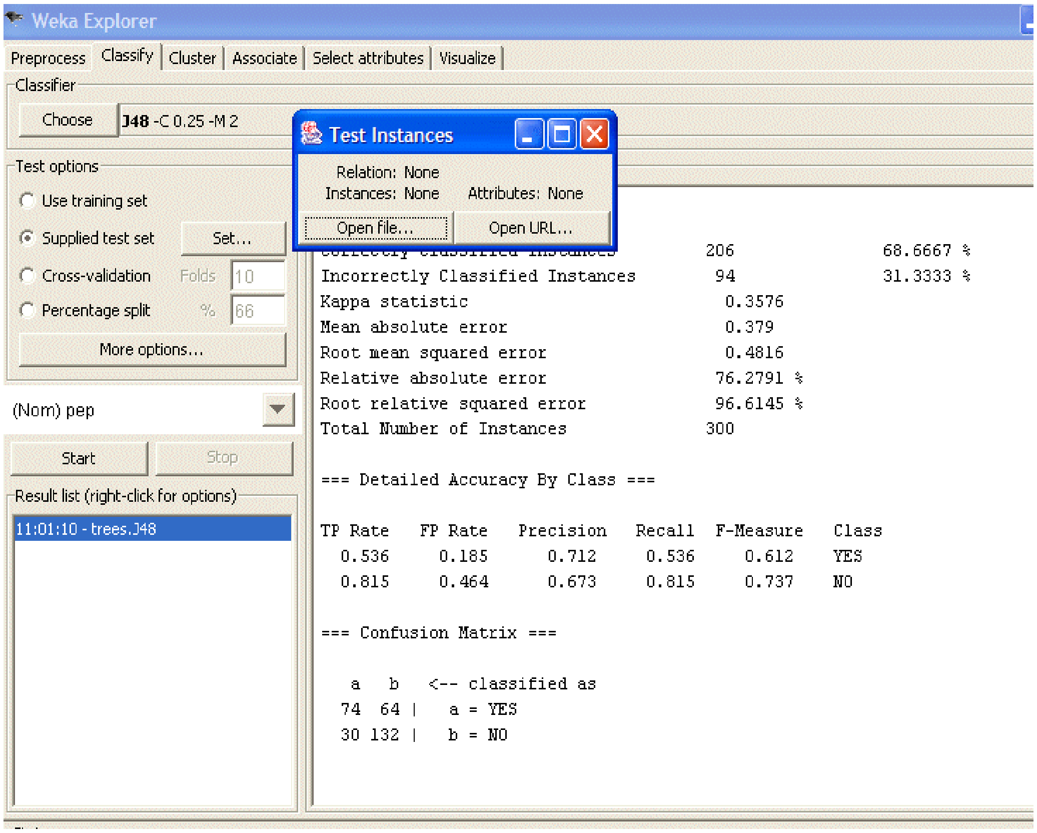


We will now use our model to classify the new instances. However, in the data section, the value of the "pep" attribute is "?" (or unknown).


TextPad - [D:\Bamshad\CLASS\ECT584\WEKA\classify\bank-new.arff]
File Edit Search View Tools Macros Configure Window Help

```
@relation bank-new.csv

@attribute age numeric
@attribute sex {MALE,FEMALE}
@attribute region {INNER_CITY,RURAL,TOWN,SUBURBAN}
@attribute income numeric
@attribute married {YES,NO}
@attribute children {YES,NO}
@attribute car {YES,NO}
@attribute mortgage {YES,NO}
@attribute pep {YES,NO}

@data

23,MALE,INNER_CITY,18766.9,YES,NO,YES,YES,?
30,MALE,RURAL,9915.67,NO,YES,NO,YES,?
45,FEMALE,RURAL,21881.6,NO,NO,YES,NO,?
50,MALE,TOWN,46794.4,YES,YES,NO,YES,?
41,FEMALE,INNER_CITY,20721.1,YES,NO,YES,NO,?
20,MALE,INNER_CITY,16688.5,NO,YES,NO,YES,?
46,FEMALE,RURAL,39068,YES,NO,YES,YES,?
50,FEMALE,INNER_CITY,27740.8,YES,YES,YES,YES,?
42,MALE,INNER_CITY,33584.9,NO,YES,YES,NO,?
57,FEMALE,TOWN,19621.3,YES,YES,YES,NO,?
63,FEMALE,INNER_CITY,47630.9,YES,NO,NO,YES,?
26,FEMALE,INNER_CITY,22378.5,NO,NO,YES,YES,?
62,FEMALE,RURAL,20837.1,YES,NO,YES,NO,?
26,FEMALE,SUBURBAN,23912.7,YES,NO,YES,NO,?
```

In the main panel, under "Test options" click the "Supplied test set" radio button, and then click the "Set..." button. This will pop up a window which allows you to open the file containing test instances.



J48 -C 0.25 -M 2

Folds: 10

%: 66

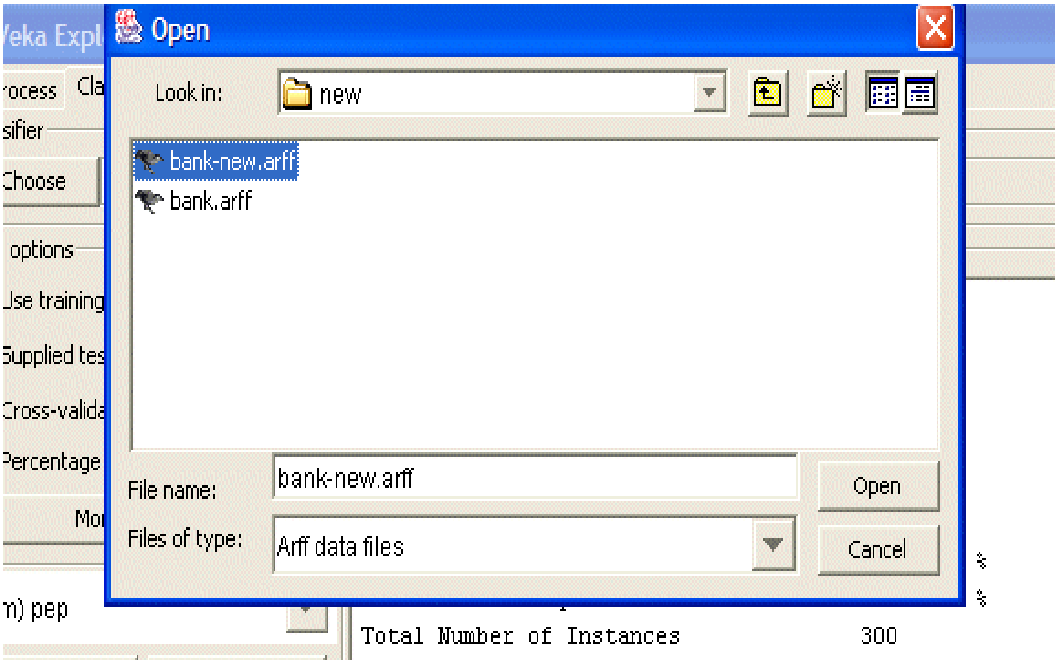
(Nom) pep

11:01:10 - trees.J48

Relation: None
Instances: None
Attributes: None

```
Correctly Classified Instances         206               68.6667 %
Incorrectly Classified Instances        94               31.3333 %
Kappa statistic                          0.3576
Mean absolute error                      0.379
Root mean squared error                  0.4816
Relative absolute error                 76.2791 %
Root relative squared error             96.6145 %
Total Number of Instances              300

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.536     0.185      0.712     0.536     0.612    YES
  0.815     0.464      0.673     0.815     0.737    NO

=== Confusion Matrix ===

   a   b   <-- classified as
  74  64 |   a = YES
  30 132 |   b = NO
```

In this case, we open the file "bank-new.arff" and upon returning to the main window, we click the "start" button. This, once again generates the models from our training data, but this time it applies the model to the new unclassified instances in the "bank-new.arff" file in order to predict the value of "pep" attribute.



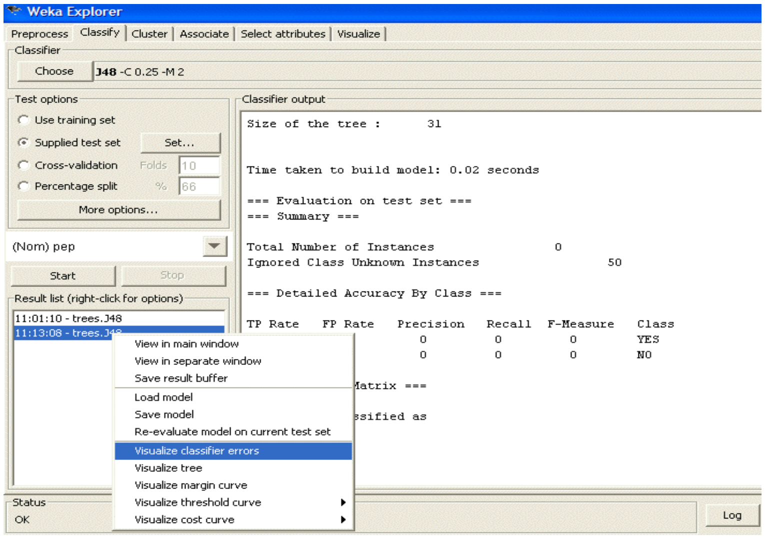
n) pep

Look in: new

File name: bank-new.arff

Files of type: Arff data files

```
Total Number of Instances              300
```



Classifier: J48 -C 0.25 -M 2

Test options: ○ Use training set ● Supplied test set ○ Cross-validation Folds 10 ○ Percentage split % 66

(Nom) pep

Result list (right-click for options)
11:01:10 - trees.J48
11:13:08 - trees.J48

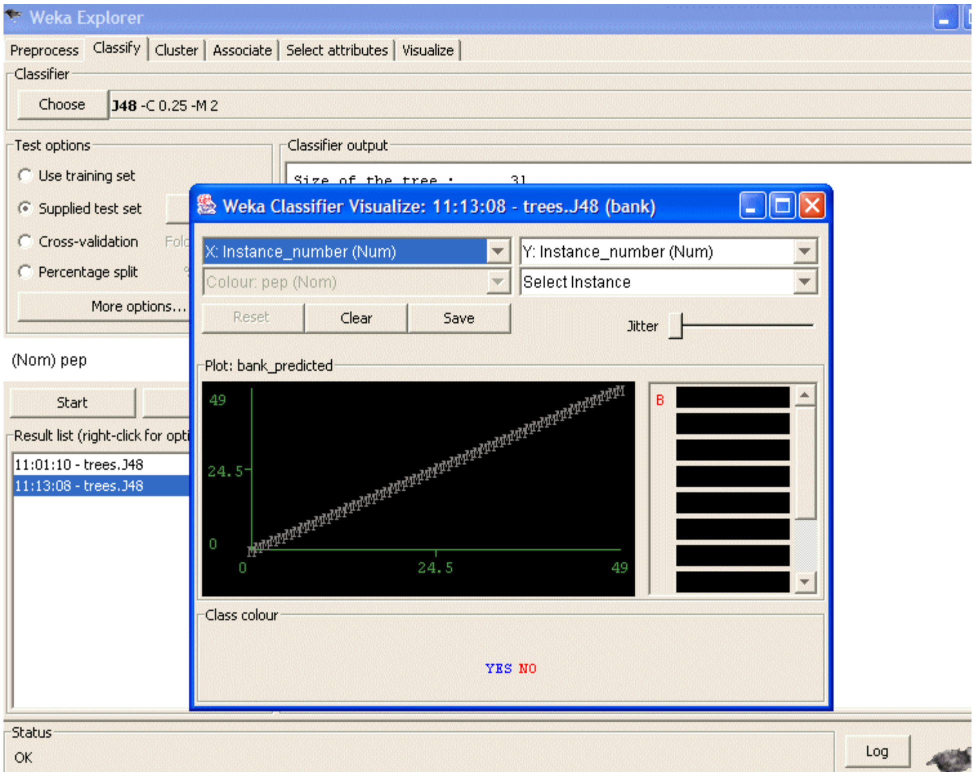
Status: OK

```
Size of the tree :      31


Time taken to build model: 0.02 seconds

=== Evaluation on test set ===
=== Summary ===

Total Number of Instances                0
Ignored Class Unknown Instances                  50

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
                        0          0         0        YES
                        0          0         0        NO

                                       Matrix ===

                                       ssified as
```

The summary of the results in the right panel does not show any statistics. This is because in our test instances the value of the class attribute ("pep") was left as "?", thus WEKA has no actual values to which it can compare the predicted values of new instances.

GUI vesion of WEKA is used to create a file containing all the new instances along with their predicted class value resulting from the application of the model.

First, right-click the most recent result set in the left "Result list" panel. In the resulting pop-up window select the menu item "Visualize classifier errors". This brings up a separate window containing a two-dimensional graph.



Classifier: J48 -C 0.25 -M 2

Test options: ○ Use training set ● Supplied test set ○ Cross-validation ○ Percentage split

(Nom) pep

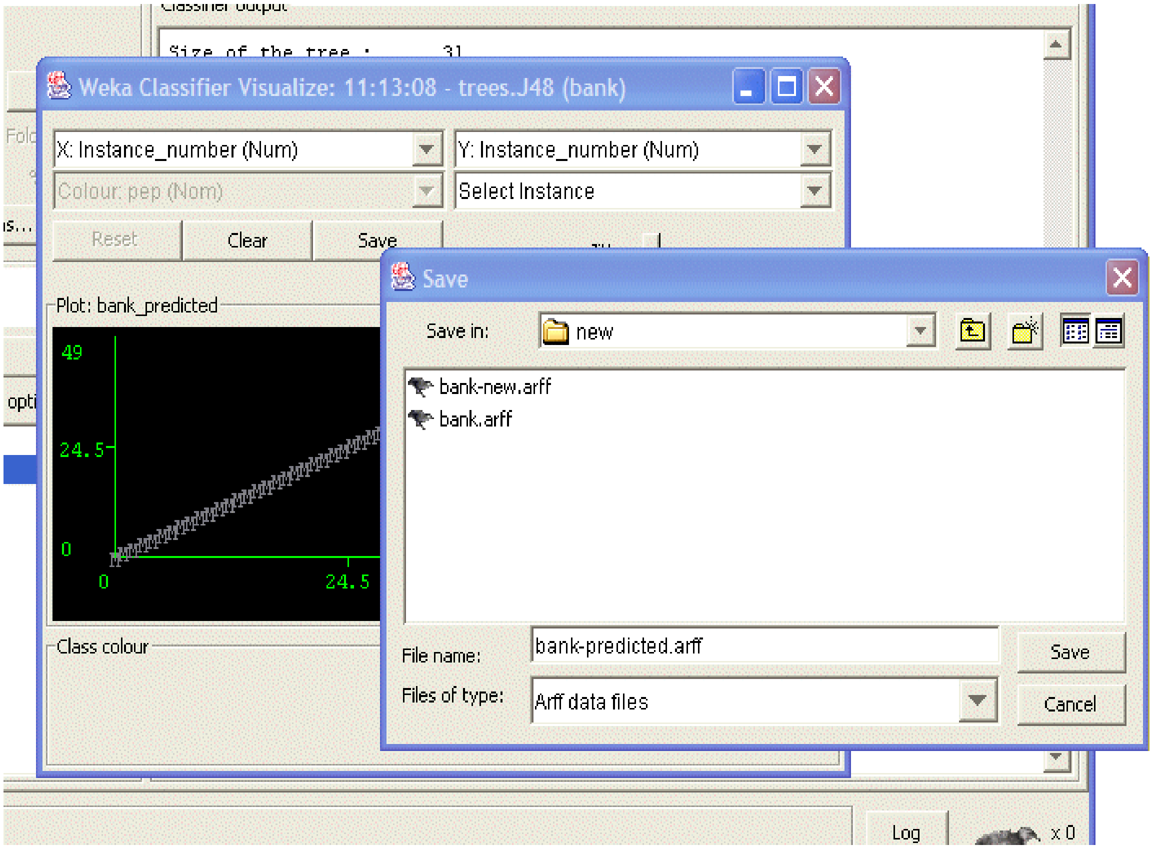
Result list (right-click for options)
11:01:10 - trees.J48
11:13:08 - trees.J48

X: Instance_number (Num)
Y: Instance_number (Num)
Colour: pep (Nom)
Select Instance

Plot: bank_predicted

Status: OK

5. To save the file: In the new window, we click on the "Save" button and save the result as the file: "bank-predicted.arff"



X: Instance_number (Num)

Y: Instance_number (Num)

Colour: pep (Nom)

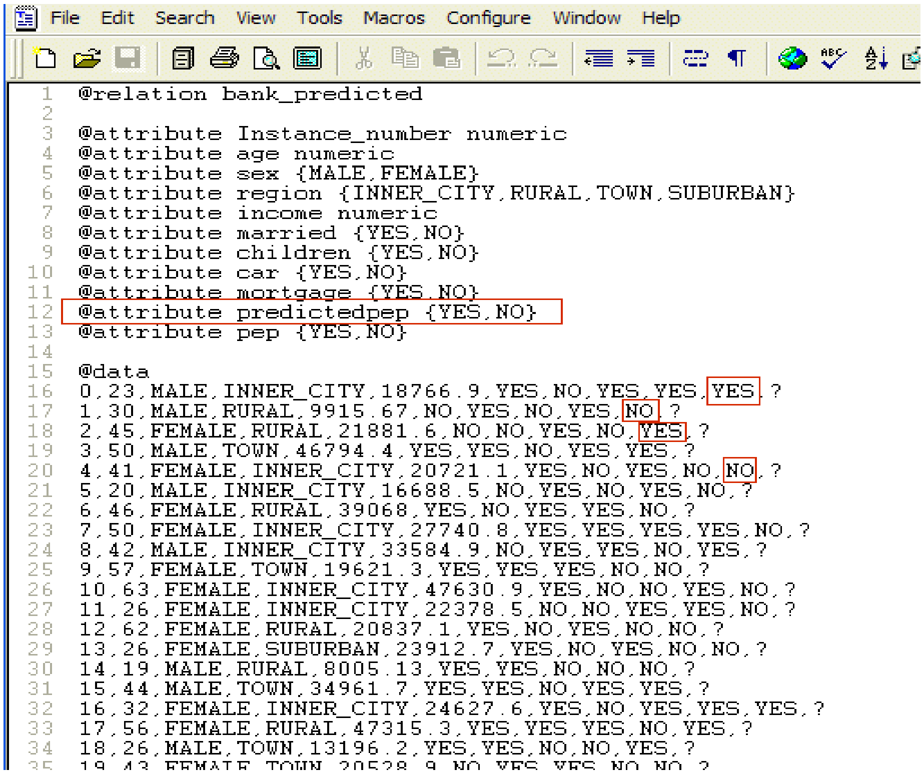
Select Instance

Save in: new

File name: bank-predicted.arff

Files of type: Arff data files

This file contains a copy of the new instances along with an additional column for the predicted value of "pep". The top portion of the file can be seen in below figure.



```
@relation bank_predicted

@attribute Instance_number numeric
@attribute age numeric
@attribute sex {MALE,FEMALE}
@attribute region {INNER_CITY,RURAL,TOWN,SUBURBAN}
@attribute income numeric
@attribute married {YES,NO}
@attribute children {YES,NO}
@attribute car {YES,NO}
@attribute mortgage {YES,NO}
@attribute predictedpep {YES,NO}
@attribute pep {YES,NO}

@data
0,23,MALE,INNER_CITY,18766.9,YES,NO,YES,YES,YES,?
1,30,MALE,RURAL,9915.67,NO,YES,NO,YES,NO,?
2,45,FEMALE,RURAL,21881.6,NO,NO,YES,NO,YES,?
3,50,MALE,TOWN,46794.4,YES,YES,NO,YES,YES,?
4,41,FEMALE,INNER_CITY,20721.1,YES,NO,YES,NO,NO,?
5,20,MALE,INNER_CITY,16688.5,NO,YES,NO,YES,NO,?
6,46,FEMALE,RURAL,39068,YES,NO,YES,YES,NO,?
7,50,FEMALE,INNER_CITY,27740.8,YES,YES,YES,YES,NO,?
8,42,MALE,INNER_CITY,33584.9,NO,YES,YES,NO,YES,?
9,57,FEMALE,TOWN,19621.3,YES,YES,YES,NO,NO,?
10,63,FEMALE,INNER_CITY,47630.9,YES,NO,NO,YES,NO,?
11,26,FEMALE,INNER_CITY,22378.5,NO,NO,YES,YES,NO,?
12,62,FEMALE,RURAL,20837.1,YES,NO,YES,NO,NO,?
13,26,FEMALE,SUBURBAN,23912.7,YES,NO,YES,NO,NO,?
14,19,MALE,RURAL,8005.13,YES,YES,NO,NO,NO,?
15,44,MALE,TOWN,34961.7,YES,YES,NO,YES,YES,?
16,32,FEMALE,INNER_CITY,24627.6,YES,NO,YES,YES,YES,?
17,56,FEMALE,RURAL,47315.3,YES,YES,YES,NO,YES,?
18,26,MALE,TOWN,13196.2,YES,YES,NO,NO,YES,?
19,43,FEMALE,TOWN,20528.9,NO,YES,YES,NO,NO,?
```

Result:

Conclusion:

The different classification algorithms of data mining were studied and one among them classification algorithm was recognized and understood like Decision tree, Naïve Bayes, Random Forest using WEKA

Industrial Application:

- OLE DB for OLAP (abbreviated ODBO) is a Microsoft published specification and an industry standard for multi-dimensional data processing. ODBO is the standard application programming interface (API) for exchanging metadata and data between an OLAP server and a client on a Windows platform.
- Marketing and sales analysis
- Consumer goods industries
- Financial services industry (insurance,banks etc.)
- Database Marketing

Questionnaires:

1. A OLAP stand for___________.
2. What are OLAP operations?
3. Data that can be modeled as dimension attributes and measure attributes are called _______ data.

A. Multidimensional B. Singledimensional C. Measured D. Dimensional

4. Compare OLTP and OLAP?
5. List the types of OLAP server.
6. Compare slicing and dicing?
7. Roll-up performs aggregation on a data cube in by climbing up a concept hierarchy for a dimension and by dimension reduction. True or False.
8. Explain Drill Down operation with suitable example.
9. A Pivot operation is also known as _______________
10. The process of viewing the cross-tab (Single dimensional) with a fixed value of one attribute is

A. Slicing B. Dicing C. Pivoting D. Both a and b

Experiment No. : 8

WEKA Tool

Aim: Implementation of Data Discretization (any one) & Visualization (any one)

Description:

1. **Visualization**

This program calculates and has comparisons on the data set selection of attributes and methods of

manipulations have been chosen. The Visualization can be shown in a 2-D representation of the information.

Creation of Weather Table:

Procedure:

1) Open Start Programs Accessories Notepad

2) Type the following training data set with the help of Notepad for Weather Table.

```
@relation weather

@attribute outlook {sunny, rainy, overcast}

@attribute temperature numeric

@attribute humidity numeric

@attribute windy {TRUE, FALSE}

@attribute play {yes, no}

@data

sunny,85,85,FALSE,no

sunny,80,90,TRUE,no

overcast,83,86,FALSE,yes

rainy,70,96,FALSE,yes

rainy,68,80,FALSE,yes

rainy,65,70,TRUE,no
```

overcast,64,65,TRUE,yes

sunny,72,95,FALSE,no

sunny,69,70,FALSE,yes

rainy,75,80,FALSE,yes

sunny,75,70,TRUE,yes

overcast,72,90,TRUE,yes

overcast,81,75,FALSE,yes

rainy,71,91,TRUE,no

3) After that the file is saved with **.arff** file format.

4) Minimize the arff file and then open Start Programs weka-3-4.

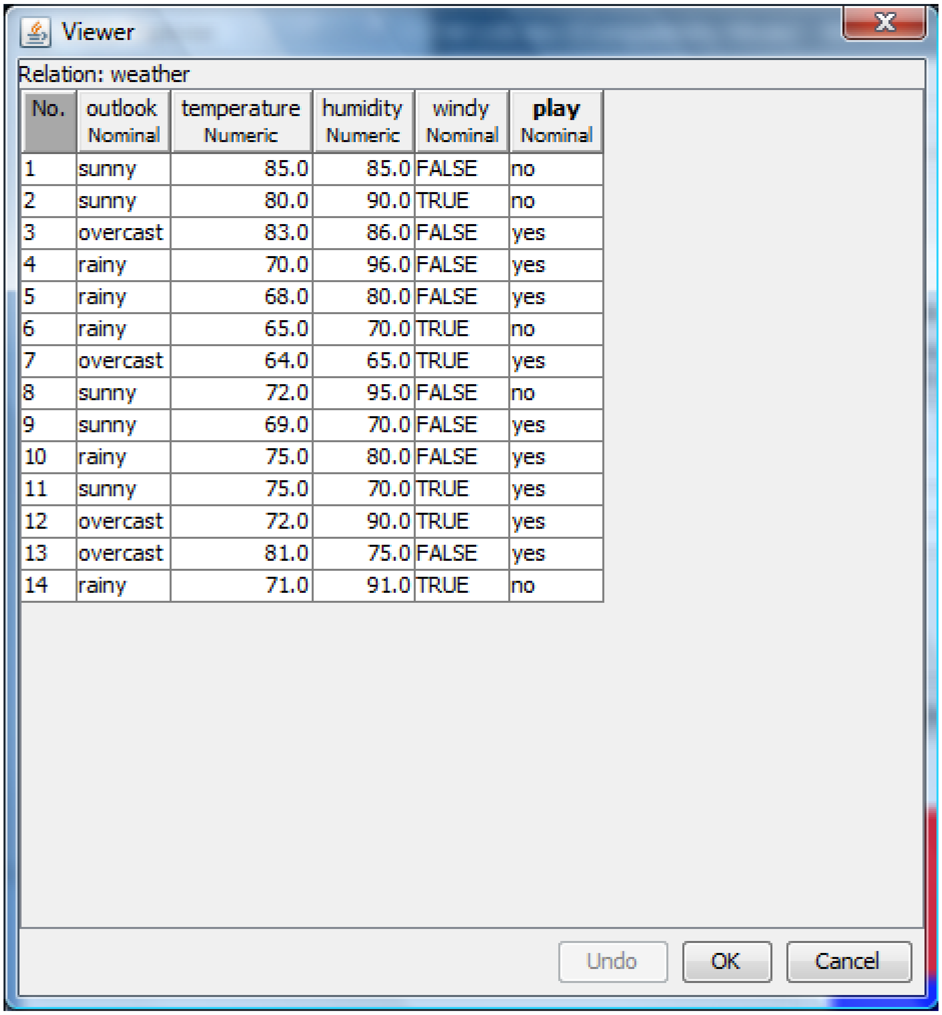
5) Click on **weka-3-4**, then Weka dialog box is displayed on the screen.

6) In that dialog box there are four modes, click on **explorer**.

7) Explorer shows many options. In that click on **‘open file’** and select the arff file

8) Click on **edit button** which shows weather table on weka.41

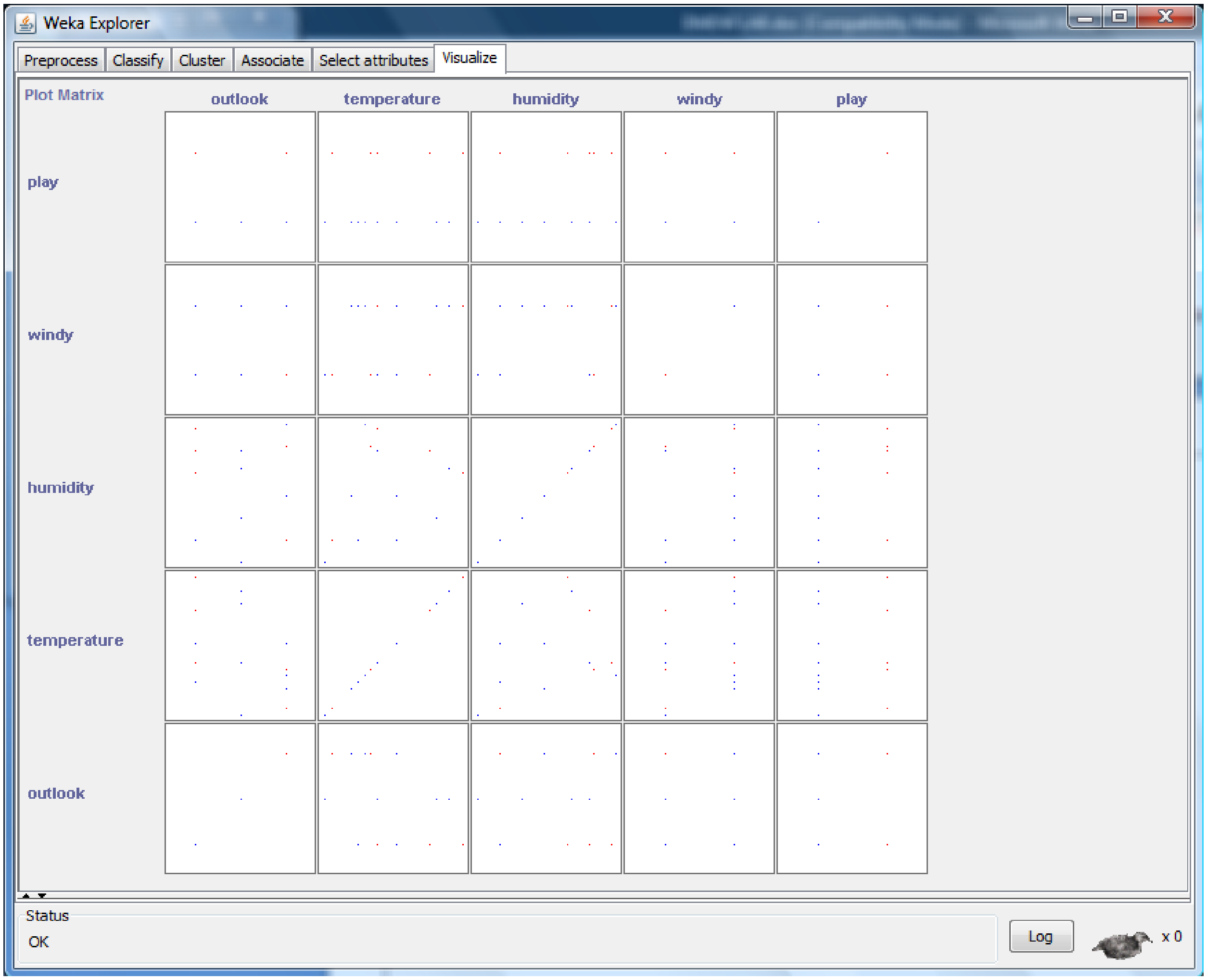
Training Data Set Weather Table



Relation: weather

No.	outlook Nominal	temperature Numeric	humidity Numeric	windy Nominal	play Nominal
1	sunny	85.0	85.0	FALSE	no
2	sunny	80.0	90.0	TRUE	no
3	overcast	83.0	86.0	FALSE	yes
4	rainy	70.0	96.0	FALSE	yes
5	rainy	68.0	80.0	FALSE	yes
6	rainy	65.0	70.0	TRUE	no
7	overcast	64.0	65.0	TRUE	yes
8	sunny	72.0	95.0	FALSE	no
9	sunny	69.0	70.0	FALSE	yes
10	rainy	75.0	80.0	FALSE	yes
11	sunny	75.0	70.0	TRUE	yes
12	overcast	72.0	90.0	TRUE	yes
13	overcast	81.0	75.0	FALSE	yes
14	rainy	71.0	91.0	TRUE	no

2-**D Plot Matrix:**42



Procedure:

1) Open Start □ Programs □ Weka-3-4 □ Weka-3-4

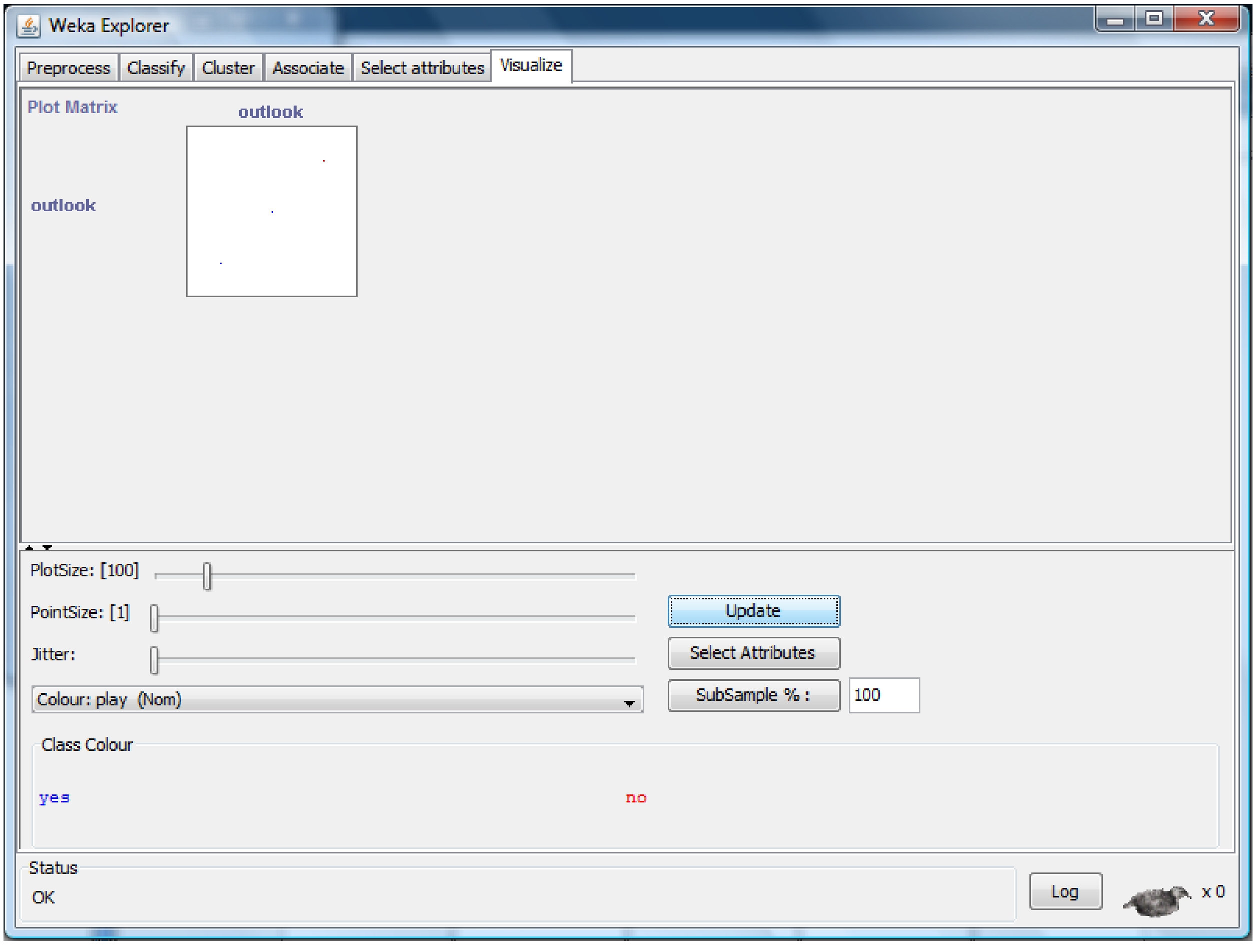
2) Open the explorer and click on **Preprocess**, then a new window will appear. In that window select

weather.arff file then the data will be displayed.

3) After that click on the **Visualize tab** on the top of the Menu bar.

4) When we select **Visualize tab** then **Plot Matrix** is displayed on the screen.

Output:



PlotSize: [100]

PointSize: [1]

Jitter:

Colour: play (Nom)

SubSample % : 100

Class Colour

yes no

Status

OK

5) After that we select the **Select Attribute button**, then select **Outlook attribute** and clock OK.

6) Click on the **Update button** to display the output.

7) After that select the **Select Attribute button** and select **Temperature attribute** and then click OK.

8) Increase the **Plot Size** and **Point Size**.

9) Click on the **Update button** to display the output.

10) After that we select the **Select Attribute button**, then select **Humidity attribute** and clock OK.

11) Click on the **Update button** to display the output.

12) After that select the **Select Attribute button** and select **Windy attribute** and then click OK.

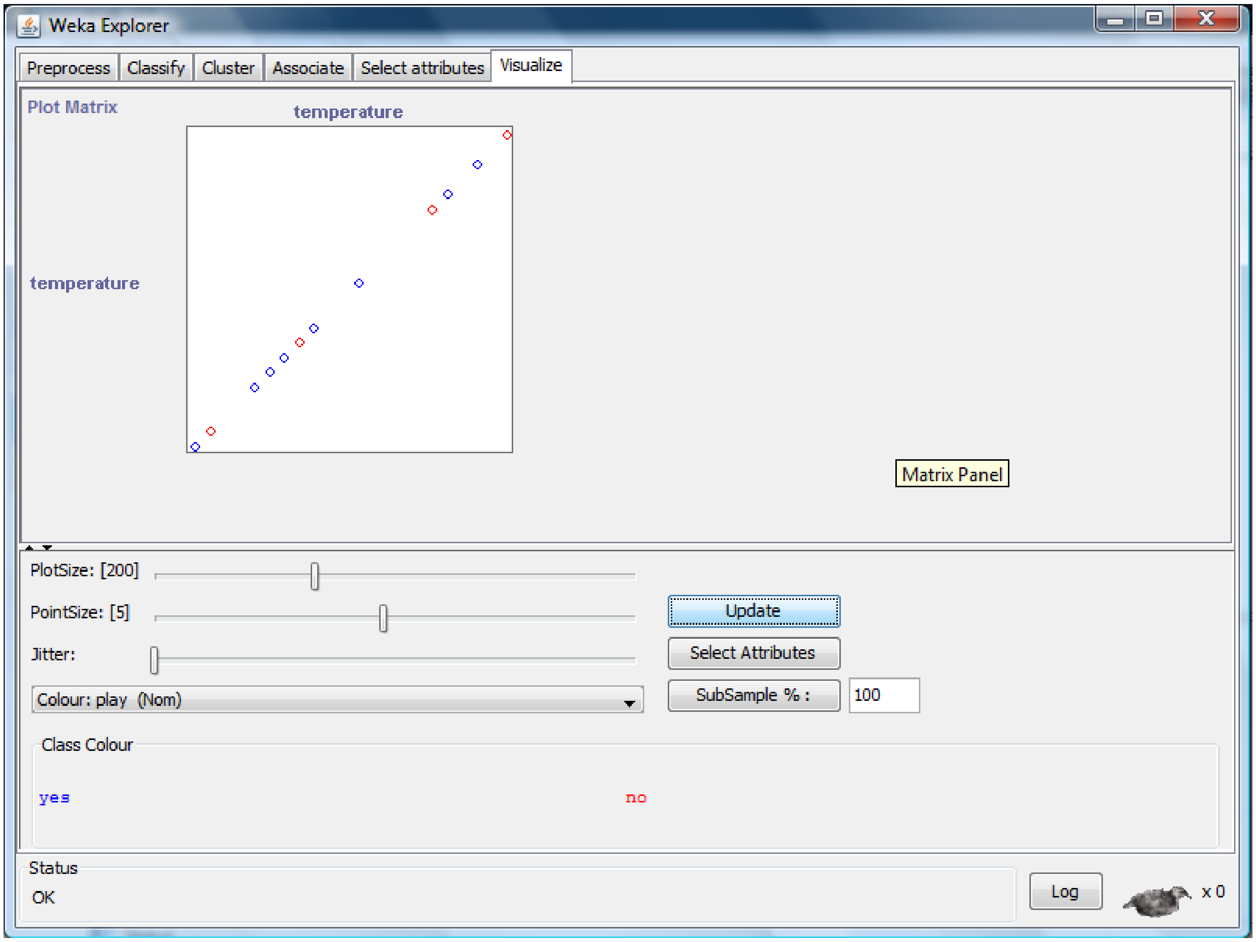
13) Increase the **Jitter Size**.

14) Click on the **Update button** to display the output.

15) After that we select the **Select Attribute button**, then select **Play attribute** and clock OK.43

16) Click on the **Update button** to display the output.

Output:



PlotSize: [200]

PointSize: [5]

Jitter:

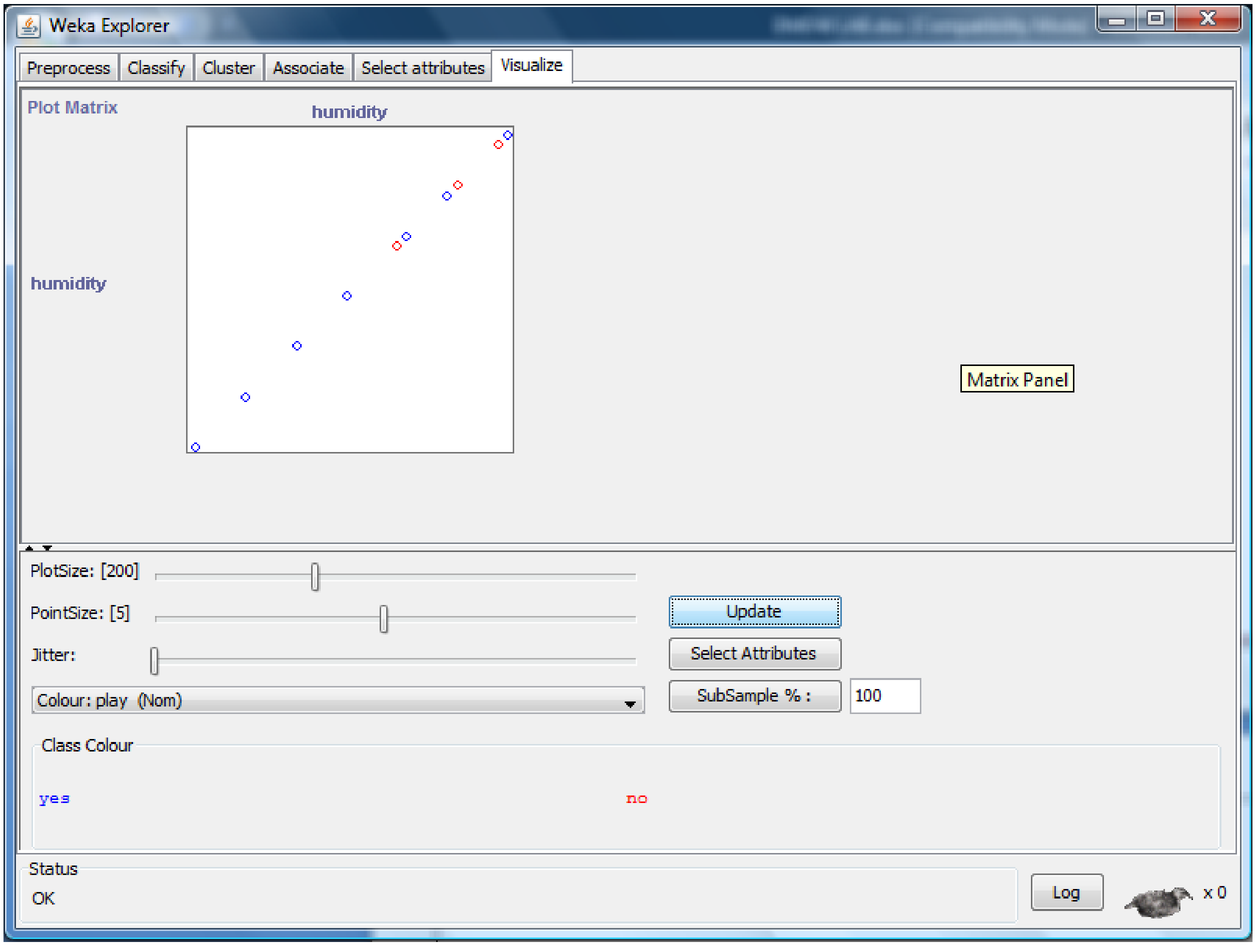
Colour: play (Nom)

SubSample % : 100

Status

OK

Output:



PlotSize: [200]

PointSize: [5]

Jitter:

Colour: play (Nom)

SubSample % : 100

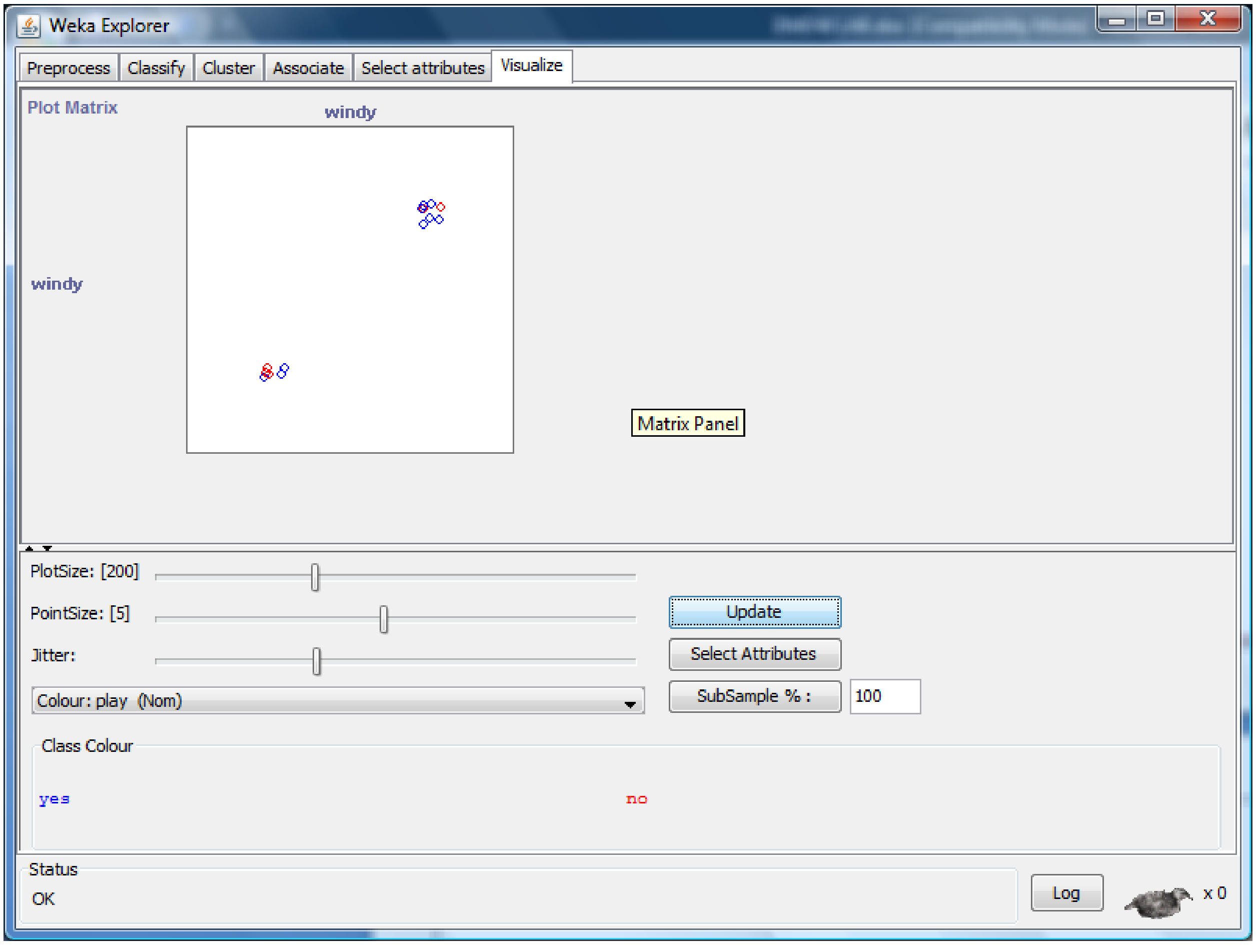
Class Colour

yes no

Status

OK

Output:



PlotSize: [200]

PointSize: [5]

Jitter:

Colour: play (Nom)

SubSample % : 100

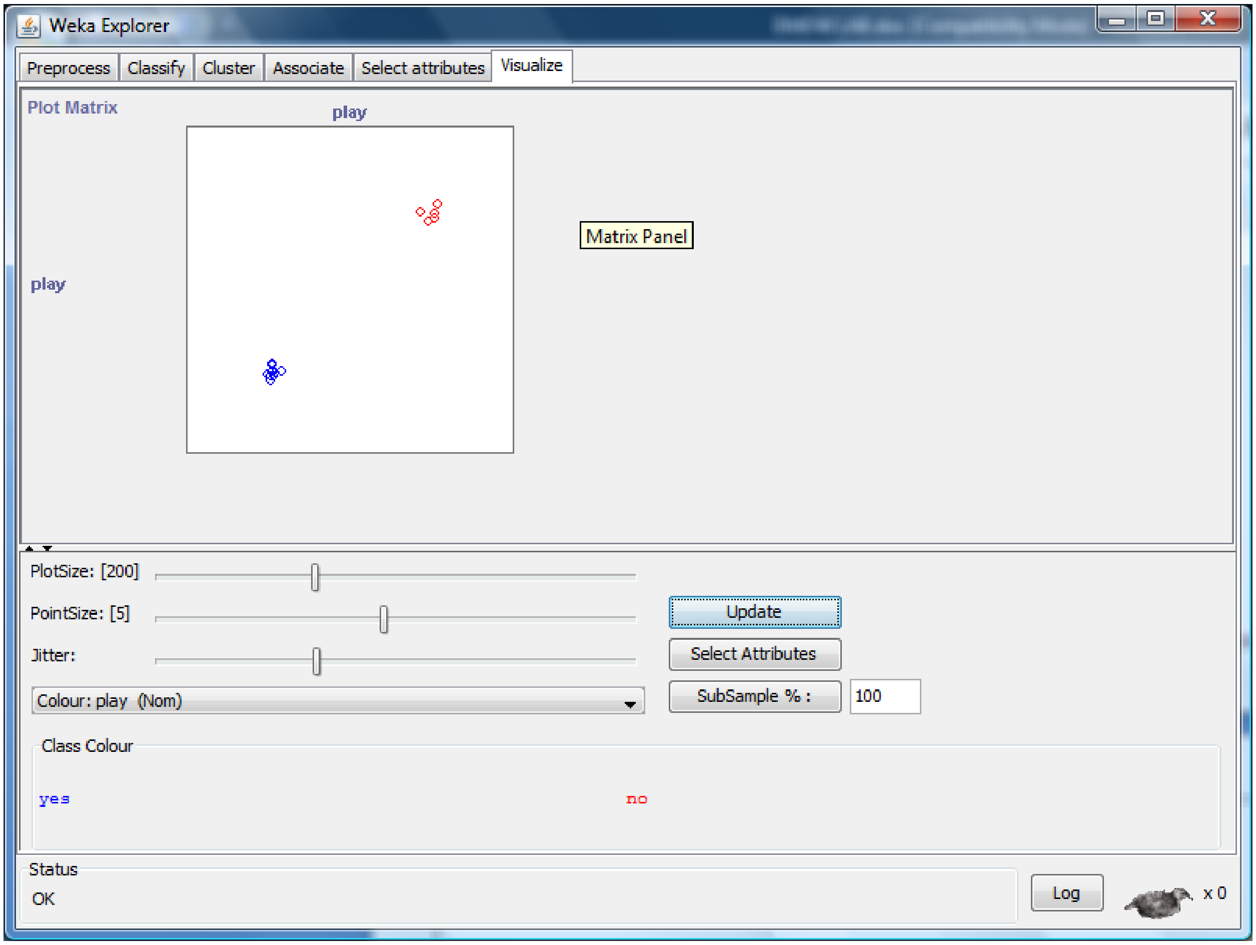
Class Colour

yes no

Status

OK

Output:



PlotSize: [200]

PointSize: [5]

Jitter:

Colour: play (Nom)

SubSample % : 100

Class Colour

yes no

Status

OK

4. Demonstration of preprocessing on dataset student.arff

This experiment illustrates some of the basic data preprocessing operations that can be performed using WEKA-Explorer. The sample dataset used for this example is the student data available in arff format.

Step1: Loading the data. We can load the dataset into weka by clicking on open button in preprocessing interface and selecting the appropriate file.

Step2: Once the data is loaded, weka will recognize the attributes and during the scan of the data weka will compute some basic strategies on each attribute. The left panel in the above figure shows the list of recognized attributes while the top panel indicates the names of the base relation or table and the current working relation (which are same initially).

Step3:Clicking on an attribute in the left panel will show the basic statistics on the attributes for the categorical attributes the frequency of each attribute value is shown, while for continuous attributes we can obtain min, max, mean, standard deviation and deviation etc., Step4:The visualization in the right button panel in the form of cross-tabulation across two attributes.

Note:we can select another attribute using the dropdown list.

Step5:Selecting or filtering attributes

Removing an attribute-When we need to remove an attribute,we can do this by using the attribute filters in weka.In the filter model panel,click on choose button,This will show a popup window with a list of available filters.

Scroll down the list and select the “weka.filters.unsupervised.attribute.remove” filters.

Step 6:a)Next click the textbox immediately to the right of the choose button.In the resulting dialog box enter the index of the attribute to be filtered out.

b)Make sure that invert selection option is set to false.The click OK now in the filter box.you will see “Remove-R-7”.

c)Click the apply button to apply filter to this data.This will remove the attribute and create new working relation.

d)Save the new working relation as an arff file by clicking save button on the top(button)panel.(student.arff)

Discretization

1)Sometimes association rule mining can only be performed on categorical data.This requires performing discretization on numeric or continuous attributes.In the following example let us discretize age attribute.

- Let us divide the values of age attribute into three bins(intervals).
- First load the dataset into weka(student.arff)
- Select the age attribute.
- Activate filter-dialog box and select “WEKA.filters.unsupervised.attribute.discretize”from

the list.

To change the defaults for the filters,click on the box immediately to the right of the choose button.

We enter the index for the attribute to be discretized.In this case the attribute is age.So we must enter ‘1’ corresponding to the age attribute.

Enter ‘3’ as the number of bins.Leave the remaining field values as they are.

Click OK button.

Click apply in the filter panel.This will result in a new working relation with the selected attribute partition into 3 bins.

Save the new working relation in a file called student-data-discretized.arff

Dataset student .arff

```
@relation student
@attribute age {<30,30-40,>40}
@attribute income {low, medium, high}
@attribute student {yes, no}
@attribute credit-rating {fair, excellent}
@attribute buyspc {yes, no}
@data
%<30, high, no, fair, no
<30, high, no, excellent, no
30-40, high, no, fair, yes
>40, medium, no, fair, yes
>40, low, yes, fair, yes
>40, low, yes, excellent, no
30-40, low, yes, excellent, yes
<30, medium, no, fair, no
```

<30, low, yes, fair, no

>40, medium, yes, fair, yes

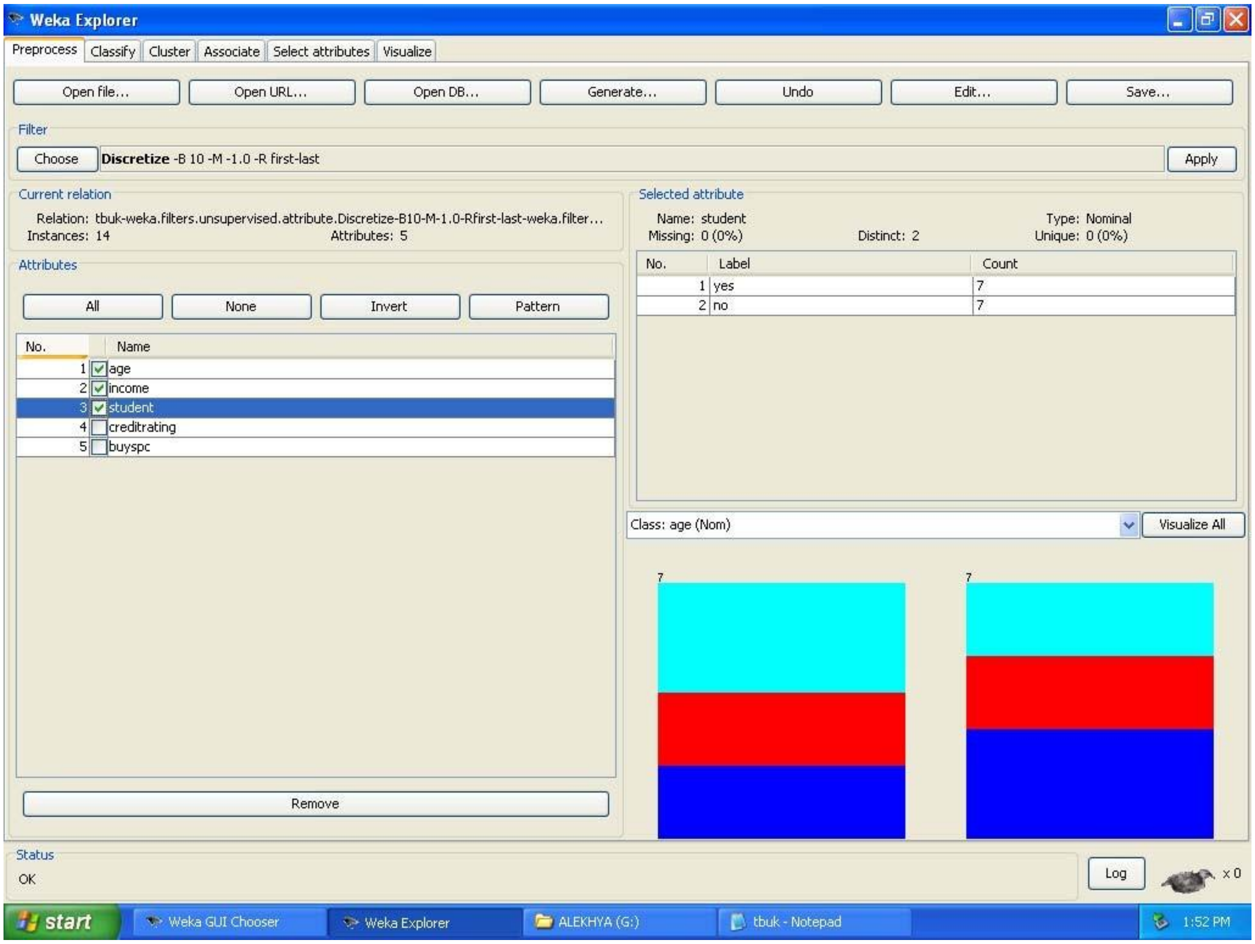
<30, medium, yes, excellent, yes

30-40, medium, no, excellent, yes

30-40, high, yes, fair, yes

>40, medium, no, excellent, no

%The following screenshot shows the effect of discretization.**2. Demonstration of preprocessing on dataset labor.arff**



Filter

Discretize -B 10 -M -1.0 -R first-last

Current relation

Relation: tbuk-weka.filters.unsupervised.attribute.Discretize-B10-M-1.0-Rfirst-last-weka.filter...
Instances: 14 Attributes: 5

Attributes

No.		Name
1	☑	age
2	☑	income
3	☑	student
4	□	creditrating
5	□	buyspc

Selected attribute

Name: student Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count
1	yes	7
2	no	7

Class: age (Nom)

Status

OK

Result:

This program has been successfully executed.

Experiment No. : 9

Clustering algorithm – K-means using WEKA tool.

Aim: To implement the following Clustering Algorithms – K-means, Agglomerative, Divisive using WEKA

Objectives: From this experiment, the student will be able to

- Analyse the data, identify the problem and choose relevant algorithm to apply
- Understand and implement classical clustering algorithms in data mining
- Identify the application of clustering algorithm in data mining

Outcomes: The learner will be able to

- Assess the strength and weaknesses of algorithms
- Identify, formulate and solve engineering problems
- Analyse the local and global impact of data mining on individuals, organizations and society

Software Required :WEKA tool

Theory:

The GUI Chooser consists of four buttons—one for each of the four major Weka applications—and four menus.The buttons can be used to start the following applications:

• Explorer : An environment for exploring data with WEKA .

• Experimenter : An environment for performing experiments and conducting statistical tests between learning schemes.

• KnowledgeFlow : This environment supports essentially the same functions as the Explorer but with a drag-and-drop interface. One advantage is that it supports incremental learning.

• SimpleCLI : Provides a simple command-line interface that allows direct execution of WEKA commands for operating systems that do not provide their own command line interface.

WEKA CLUSTERER

It contains "clusterers" for finding groups of similar instances in a dataset. Some implemented schemes are: *k*-Means, EM, Cobweb, *X*-means, FarthestFirst .Clusters can be visualized and compared to "true" clusters.

1. Procedure:

The basic step of k-means clustering is simple. In the beginning, we determine number of cluster K and we assume the centroid or center of these clusters. We can take any random objects as the initial centroids or the first K objects can also serve as the initial centroids. Then the K means algorithm will do the three steps below until convergence. Iterate until *stable* (= no object move group):

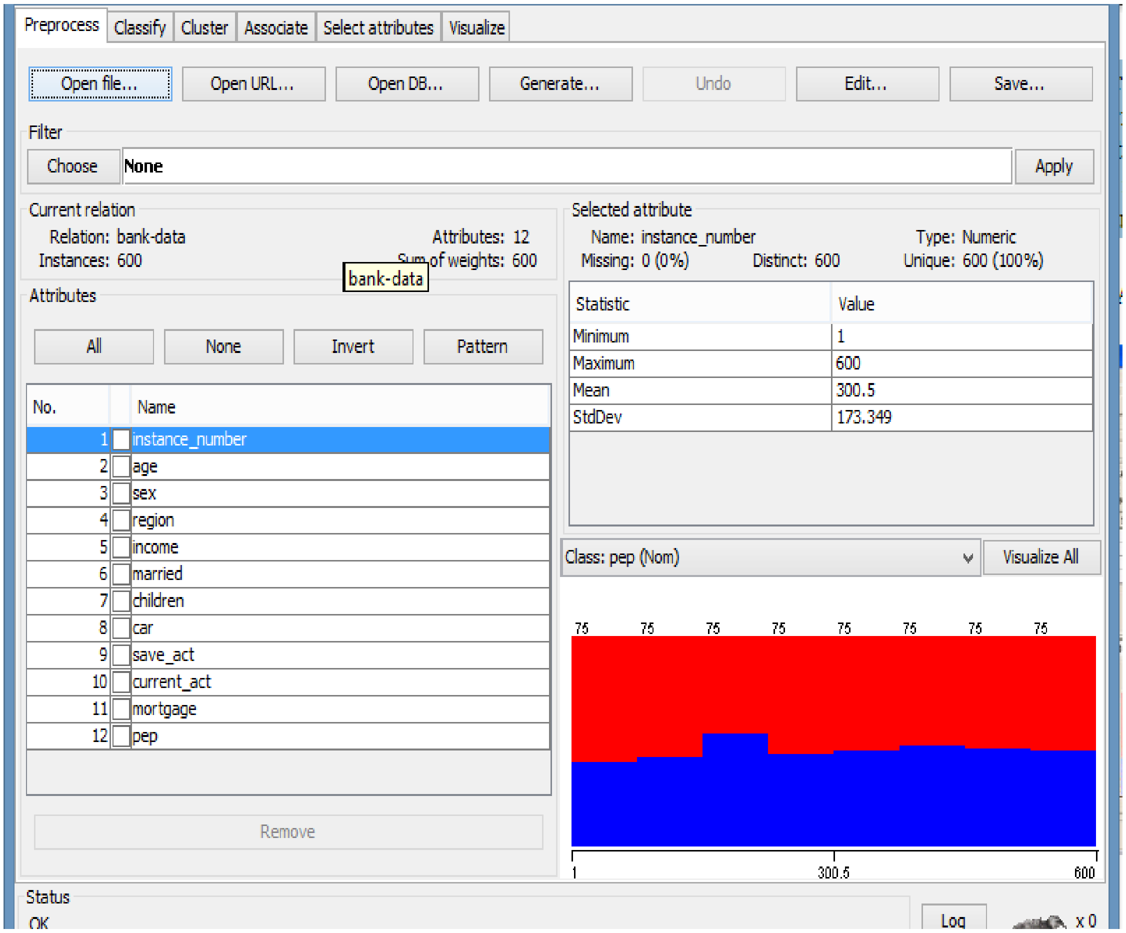
1. Determine the centroid coordinate

2. Determine the distance of each object to the centroids

3. Group the object based on minimum distance (find the closest centroid)

K-means in WEKA 3.7

The sample data set used is based on the "bank data" available in comma-separated format bank-data.csv. The resulting data file is "bank.arff" and includes 600 instances. As an illustration of performing clustering in WEKA, we will use its implementation of the K-means algorithm to cluster the cutomers in this bank data set, and to characterize the resulting customer segments.



Filter: None

Current relation
Relation: bank-data
Instances: 600
Attributes: 12
Sum of weights: 600

Selected attribute
Name: instance_number
Type: Numeric
Missing: 0 (0%)
Distinct: 600
Unique: 600 (100%)

Statistic	Value
Minimum	1
Maximum	600
Mean	300.5
StdDev	173.349

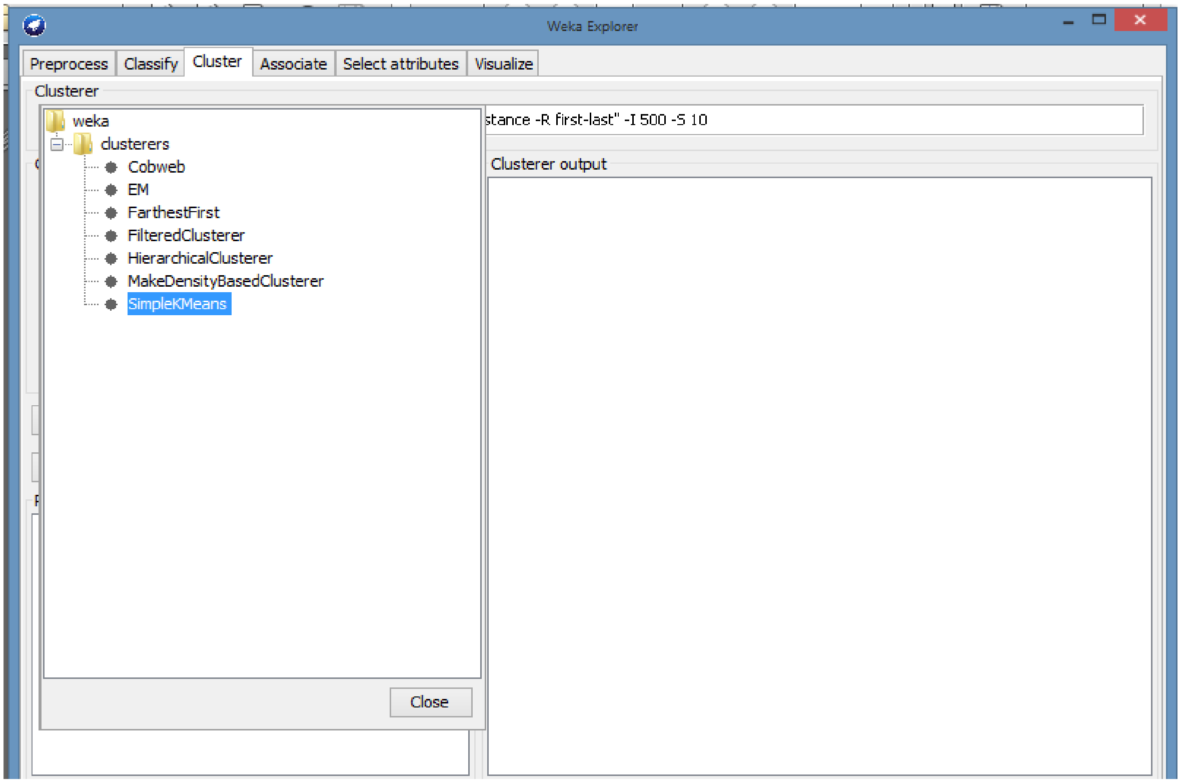
Class: pep (Nom)

Attributes

No.	Name
1	instance_number
2	age
3	sex
4	region
5	income
6	married
7	children
8	car
9	save_act
10	current_act
11	mortgage
12	pep

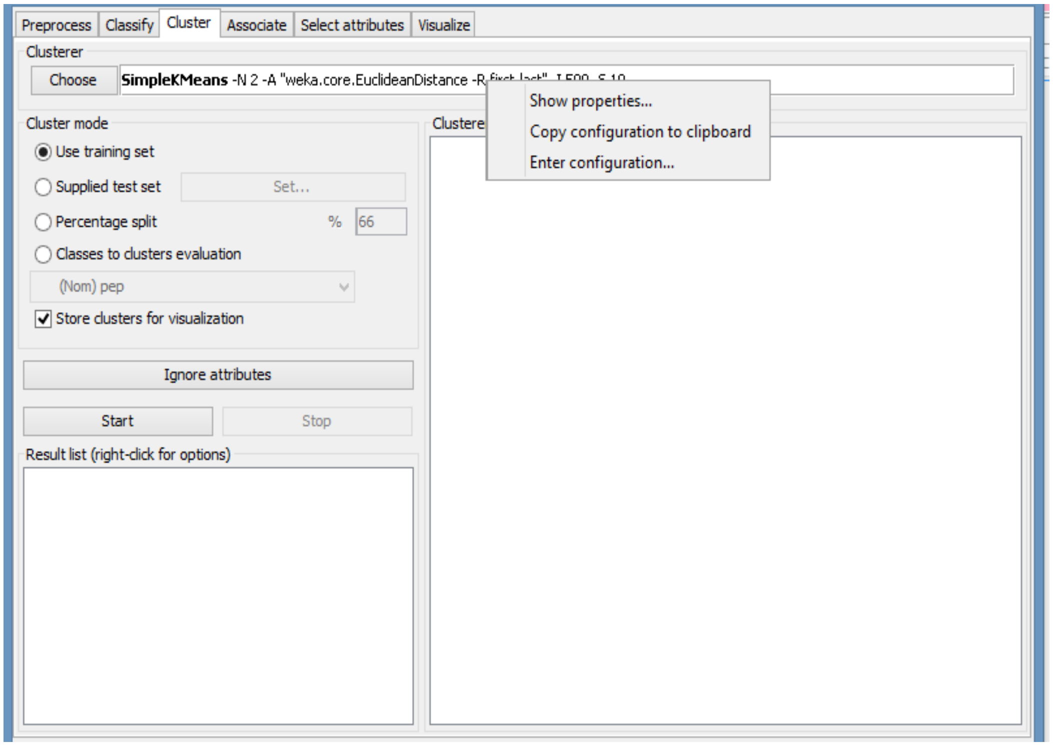
Status
OK

To perform clustering, select the "Cluster" tab in the Explorer and click on the "Choose" button. This results in a drop down list of available clustering algorithms. In this case we select "SimpleKMeans".



stance -R first-last" -I 500 -S 10

Next, click on the text box to the right of the "Choose" button to get the pop-up window shown below, for editing the clustering parameter.



SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10

● Use training set

○ Supplied test set

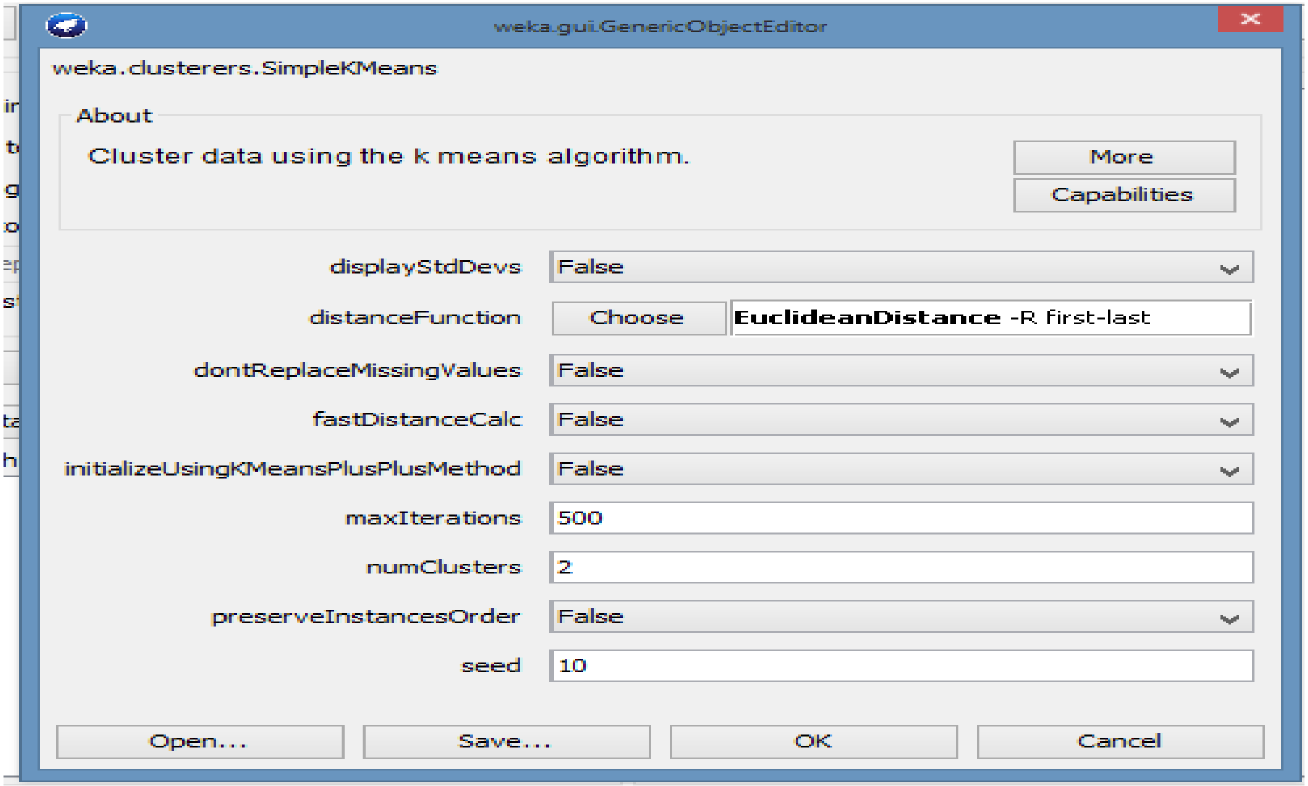
○ Percentage split % 66

○ Classes to clusters evaluation

(Nom) pep

☑ Store clusters for visualization

In the pop-up window we enter **2** as the number of clusters and we leave the value of "seed" as is



weka.gui.GenericObjectEditor

weka.clusterers.SimpleKMeans

About

Cluster data using the k means algorithm.

displayStdDevs: False

distanceFunction: Choose **EuclideanDistance** -R first-last

dontReplaceMissingValues: False

fastDistanceCalc: False

initializeUsingKMeansPlusPlusMethod: False

maxIterations: 500

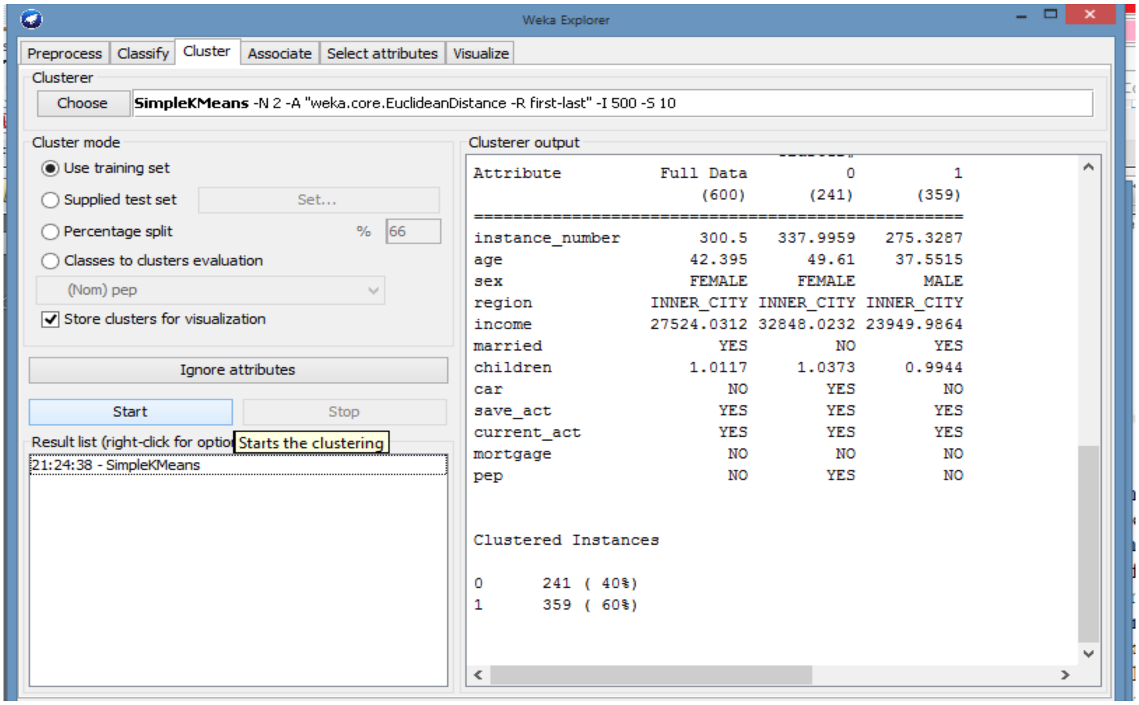
numClusters: 2

preserveInstancesOrder: False

seed: 10

The seed value is used in generating a random number which is, in turn, used for making the initial assignment of instances to clusters.

Once the options have been specified, we can run the clustering algorithm. Here we make sure that in the "Cluster Mode" panel, the "Use training set" option is selected, and we click "Start".



Weka Explorer

Clusterer

Choose **SimpleKMeans** -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10

Cluster mode

● Use training set
○ Supplied test set Set...
○ Percentage split % 66
○ Classes to clusters evaluation
(Nom) pep
☑ Store clusters for visualization

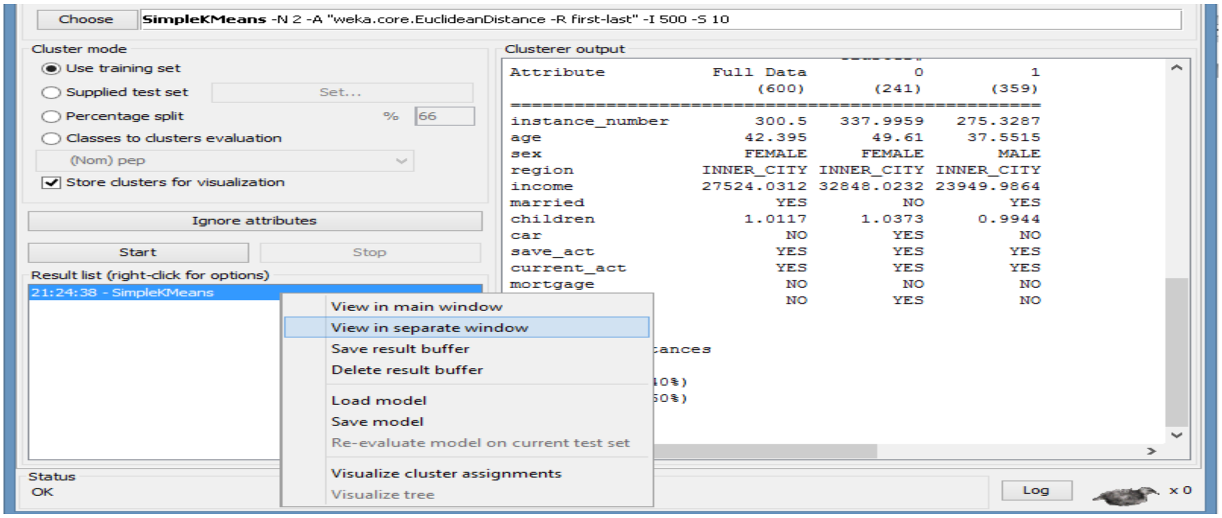
Result list (right-click for optio

21:24:38 - SimpleKMeans

Clusterer output

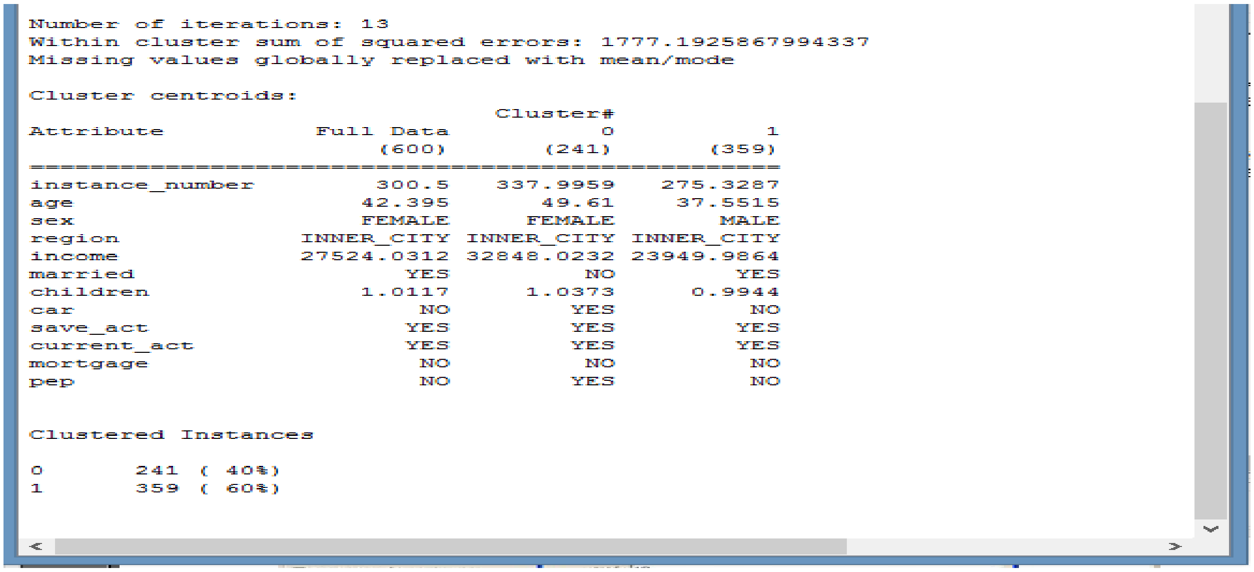
```
Attribute           Full Data          0          1
                       (600)      (241)      (359)
===================================================
instance_number        300.5   337.9959   275.3287
age                   42.395      49.61    37.5515
sex                   FEMALE     FEMALE       MALE
region            INNER_CITY INNER_CITY INNER_CITY
income            27524.0312 32848.0232 23949.9864
married                  YES         NO        YES
children              1.0117     1.0373     0.9944
car                       NO        YES         NO
save_act                 YES        YES        YES
current_act              YES        YES        YES
mortgage                  NO         NO         NO
pep                       NO        YES         NO


Clustered Instances

0       241 ( 40%)
1       359 ( 60%)
```

We can right click the result set in the "Result list" panel and view the results of clustering in a separate window.



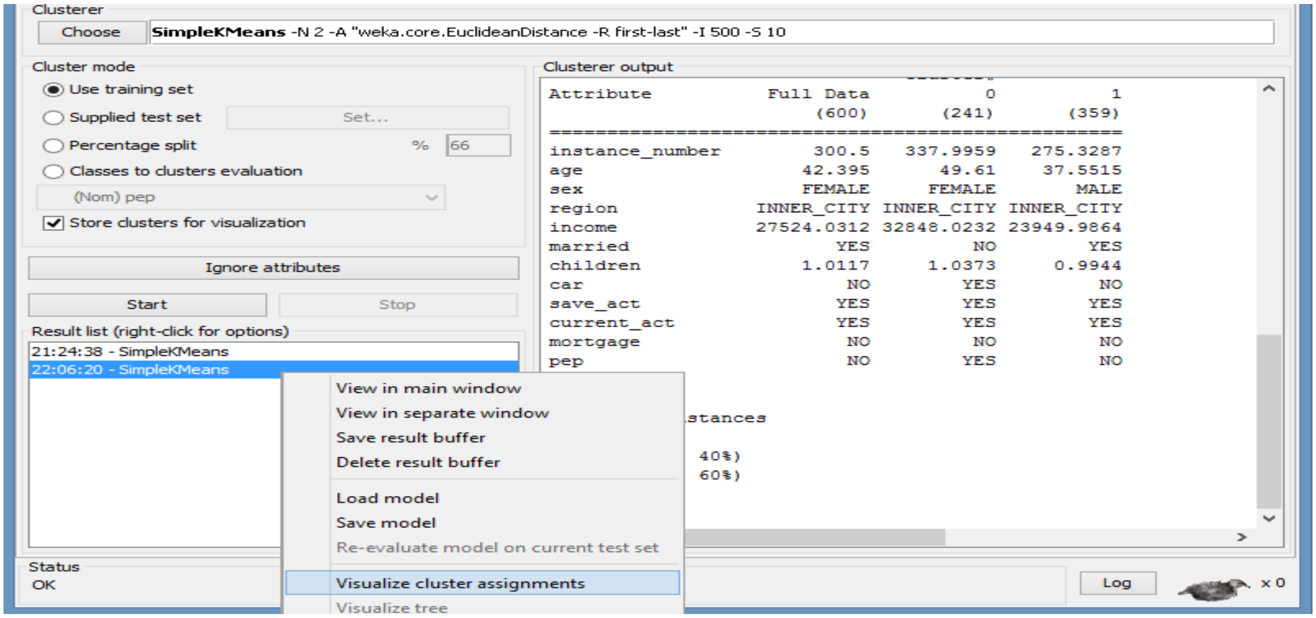
SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10

```
Attribute          Full Data         0          1
                       (600)     (241)      (359)
==================================================
instance_number        300.5  337.9959   275.3287
age                   42.395     49.61    37.5515
sex                   FEMALE    FEMALE       MALE
region            INNER_CITY INNER_CITY INNER_CITY
income            27524.0312 32848.0232 23949.9864
married                  YES        NO        YES
children              1.0117    1.0373     0.9944
car                       NO       YES         NO
save_act                 YES       YES        YES
current_act              YES       YES        YES
mortgage                  NO        NO         NO
                          NO       YES         NO
```



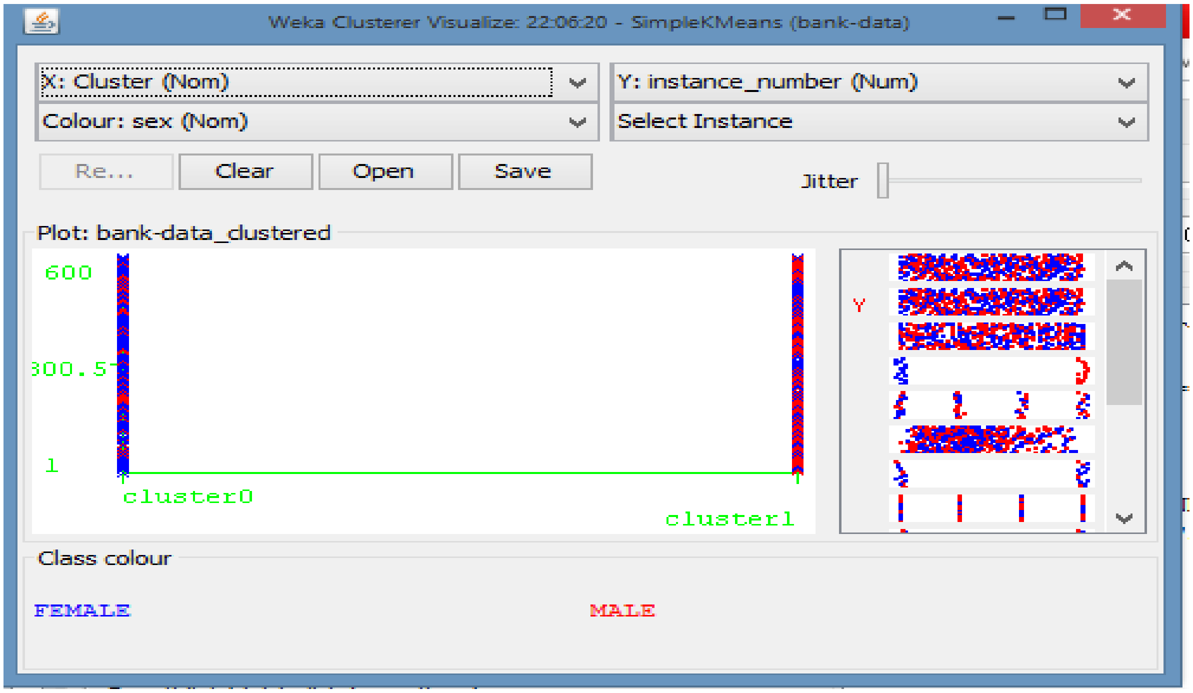
```
Number of iterations: 13
Within cluster sum of squared errors: 1777.1925867994337
Missing values globally replaced with mean/mode

Cluster centroids:
                                     Cluster#
Attribute                Full Data          0          1
                             (600)      (241)      (359)
=========================================================
instance_number              300.5   337.9959   275.3287
age                         42.395      49.61    37.5515
sex                         FEMALE     FEMALE       MALE
region                  INNER_CITY INNER_CITY INNER_CITY
income                  27524.0312 32848.0232 23949.9864
married                        YES         NO        YES
children                    1.0117     1.0373     0.9944
car                             NO        YES         NO
save_act                       YES        YES        YES
current_act                    YES        YES        YES
mortgage                        NO         NO         NO
pep                             NO        YES         NO


Clustered Instances

0       241 ( 40%)
1       359 ( 60%)
```

We can even visualize the assigned cluster as below



SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10

```
Attribute          Full Data         0          1
                       (600)     (241)      (359)
==================================================
instance_number        300.5  337.9959   275.3287
age                   42.395     49.61    37.5515
sex                   FEMALE    FEMALE       MALE
region            INNER_CITY INNER_CITY INNER_CITY
income            27524.0312 32848.0232 23949.9864
married                  YES        NO        YES
children              1.0117    1.0373     0.9944
car                       NO       YES         NO
save_act                 YES       YES        YES
current_act              YES       YES        YES
mortgage                  NO        NO         NO
pep                       NO       YES         NO

stances

 40%)
 60%)
```



Weka Clusterer Visualize: 22:06:20 - SimpleKMeans (bank-data)
Re...
Clear
Open
Save
Jitter
Plot: bank-data_clustered
600
300.5
1
cluster0
cluster1
Y
Class colour
FEMALE
MALE

X: Cluster (Nom)

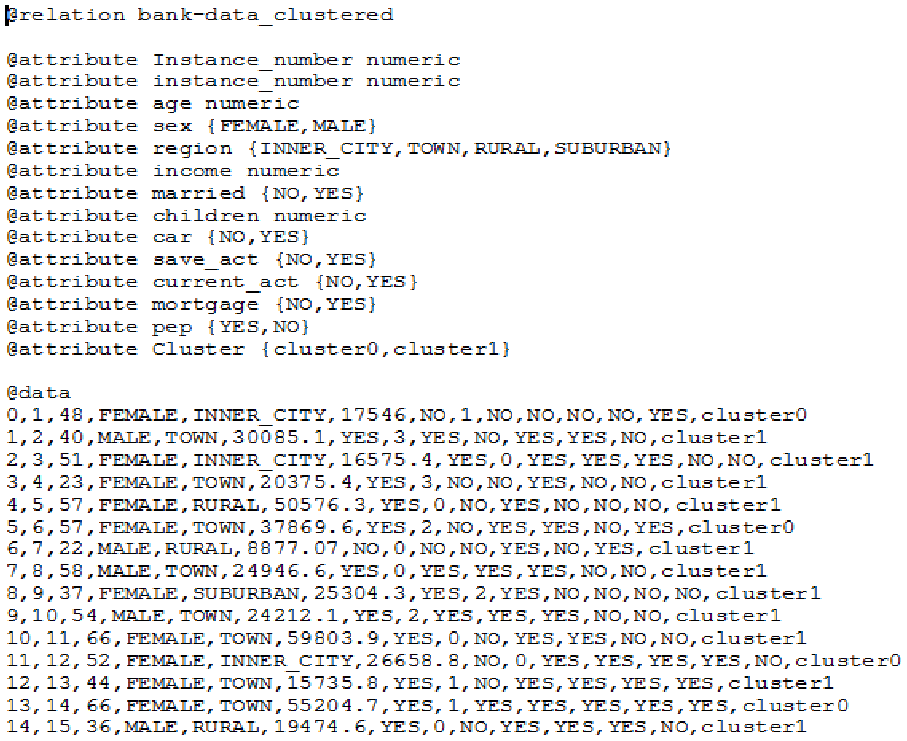
Y: instance_number (Num)

Colour: sex (Nom)

Select Instance

You can choose the cluster number and any of the other attributes for each of the three different dimensions available (x-axis, y-axis, and color). Different combinations of choices will result in a visual rendering of different relationships within each cluster.

Note that in addition to the "instance_number" attribute, WEKA has also added "Cluster" attribute to the original data set. In the data portion, each instance now has its assigned cluster as the last attribute value (as shown below).



```
@relation bank-data_clustered

@attribute Instance_number numeric
@attribute instance_number numeric
@attribute age numeric
@attribute sex {FEMALE,MALE}
@attribute region {INNER_CITY,TOWN,RURAL,SUBURBAN}
@attribute income numeric
@attribute married {NO,YES}
@attribute children numeric
@attribute car {NO,YES}
@attribute save_act {NO,YES}
@attribute current_act {NO,YES}
@attribute mortgage {NO,YES}
@attribute pep {YES,NO}
@attribute Cluster {cluster0,cluster1}

@data
0,1,48,FEMALE,INNER_CITY,17546,NO,1,NO,NO,NO,NO,YES,cluster0
1,2,40,MALE,TOWN,30085.1,YES,3,YES,NO,YES,YES,NO,cluster1
2,3,51,FEMALE,INNER_CITY,16575.4,YES,0,YES,YES,YES,NO,NO,cluster1
3,4,23,FEMALE,TOWN,20375.4,YES,3,NO,NO,YES,NO,NO,cluster1
4,5,57,FEMALE,RURAL,50576.3,YES,0,NO,YES,NO,NO,NO,cluster1
5,6,57,FEMALE,TOWN,37869.6,YES,2,NO,YES,YES,NO,YES,cluster0
6,7,22,MALE,RURAL,8877.07,NO,0,NO,NO,YES,NO,YES,cluster1
7,8,58,MALE,TOWN,24946.6,YES,0,YES,YES,YES,NO,NO,cluster1
8,9,37,FEMALE,SUBURBAN,25304.3,YES,2,YES,NO,NO,NO,NO,cluster1
9,10,54,MALE,TOWN,24212.1,YES,2,YES,YES,YES,NO,NO,cluster1
10,11,66,FEMALE,TOWN,59803.9,YES,0,NO,YES,YES,NO,NO,cluster1
11,12,52,FEMALE,INNER_CITY,26658.8,NO,0,YES,YES,YES,YES,NO,cluster0
12,13,44,FEMALE,TOWN,15735.8,YES,1,NO,YES,YES,YES,YES,cluster1
13,14,66,FEMALE,TOWN,55204.7,YES,1,YES,YES,YES,YES,YES,cluster0
14,15,36,MALE,RURAL,19474.6,YES,0,NO,YES,YES,YES,NO,cluster1
```

Result:

Conclusion:

The different clustering algorithms of data mining were studied and one among them named k-means clustering algorithm was implemented using JAVA. The need for clustering algorithm was recognized and understood.

Industrial Application:

- Anomaly or Fraud Detection:Separate valid activity groups from bots detect fraudulent claims.
- Inventory Categorization based on sales or other manufacturing metrics
- Creating NewsFeeds: K-Means can be used to cluster articles by their similarity — it can separate documents into disjoint clusters.
- Cloud Computing Environment: Clustered storage to increase performance, capacity, or reliability — clustering distributes work loads to each server, manages the transfer of workloads between servers, and provides access to all files from any server regardless of the physical location of the file.
- Environmental risks: K-means can be used to analyze environmental risk in an area — environmental risk zoning of a chemical industrial area.
- Pattern Recognition in images: For example, to automatically detect infected fruits or for segmentation of blood cells for leukemia detection.

Questionaries:

1. Define K-mean clustering.
2. What are different clustering techniques?
3. Compare K-means and K-medoids?
4. What is dendogram?
5. The dendrogram is read from right to left.True or False?
6. Which method of analysis does not classify variables as dependent or independent?

   a. regression analysis
   b. discriminant analysis
   c. analysis of variance
   d. cluster analysis

7._____ is frequently referred to as *k*-means clustering.

   a. Non-hierarchical clustering
   b. Optimizing partitioning
   c. Divisive clustering
   d. Agglomerative clustering

8.The most important part of _____ is selecting the variables on which clustering is based.

   a. interpreting and profiling clusters
   b. selecting a clustering procedure

c. assessing the validity of clustering
d. formulating the clustering problem

9. Can decision trees be used for performing clustering?Y

10.State the advantages of K-mean clustering.

Experiment No. : 10

Association Mining like Apriori, FPM.

Aim: Implementation of Apriori algorithm in WEKA.

Objectives: From this experiment, the student will be able to

- Analyse the data, identify the problem and choose relevant algorithm to apply
- Understand and implement classical association mining algorithms
- Identify the application of association mining algorithms

Outcomes: The learner will be able to

- Assess the strength and weaknesses of algorithms
- Identify, formulate and solve engineering problems
- Analyse the local and global impact of data mining on individuals, organizations and society

Software Required :WEKA tool

Theory:

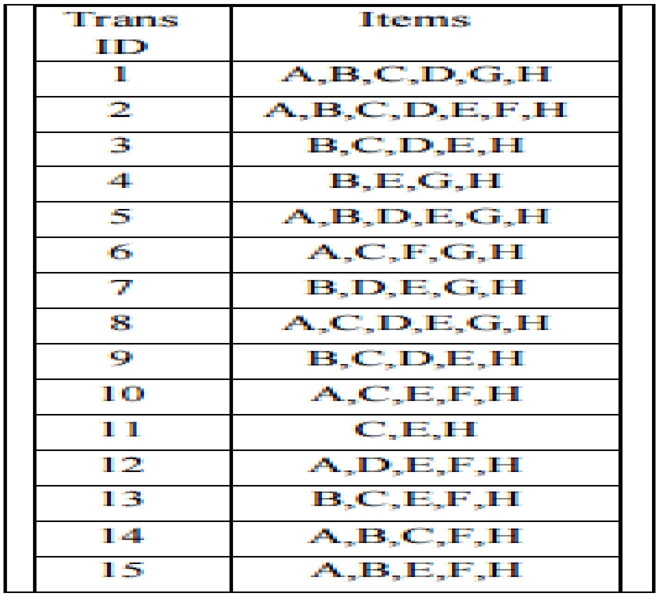
The Apriori Algorithm is an influential algorithm for mining frequent itemsets for boolean association rules. Some key concepts for Apriori algorithm are:

- Frequent Item sets: The sets of item which has minimum support (denoted by Li for ith-Itemset)
- Apriori Property: Any subset of frequent item set must be frequent.
- Join Operation: To find Lk , a set of candidate k itemsets is generated by joining Lk-1 with itself.

Procedure:

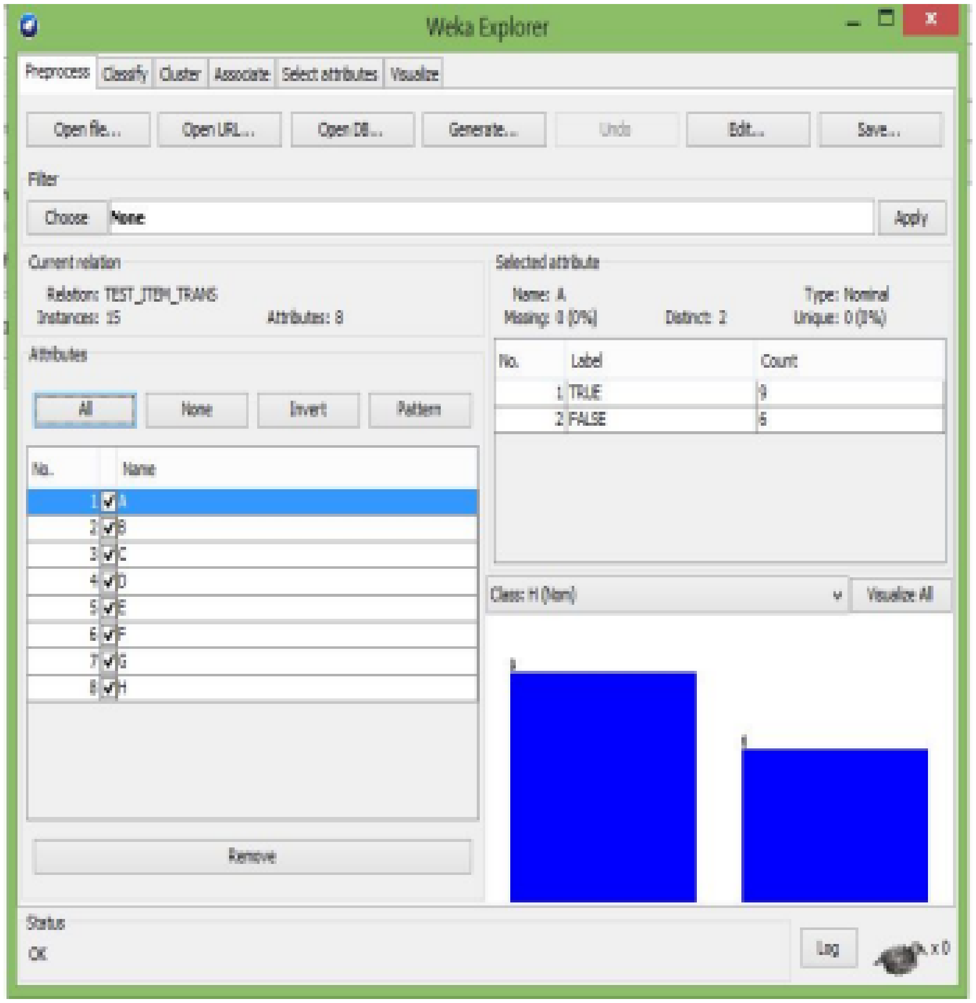
WEKA implementation:

To learn the system, TEST_ITEM_TRANS.arff has been used.



Trans ID	Items
1	A,B,C,D,G,H
2	A,B,C,D,E,F,H
3	B,C,D,E,H
4	B,E,G,H
5	A,B,D,E,G,H
6	A,C,F,G,H
7	B,D,E,G,H
8	A,C,D,E,G,H
9	B,C,D,E,H
10	A,C,E,F,H
11	C,E,H
12	A,D,E,F,H
13	B,C,E,F,H
14	A,B,C,F,H
15	A,B,E,F,H

Using the Apriori Algorithm we want to find the association rules that have minSupport=50% and minimum confidence=50%. After we launch the WEKA application and open the TEST_ITEM_TRANS.arff file as shown in below figure.



Filter: None

Current relation
Relation: TEST_ITEM_TRANS
Instances: 15
Attributes: 8

Selected attribute
Name: A
Type: Nominal
Missing: 0 (0%)
Distinct: 2
Unique: 0 (0%)

No.	Label	Count
1	TRUE	9
2	FALSE	6

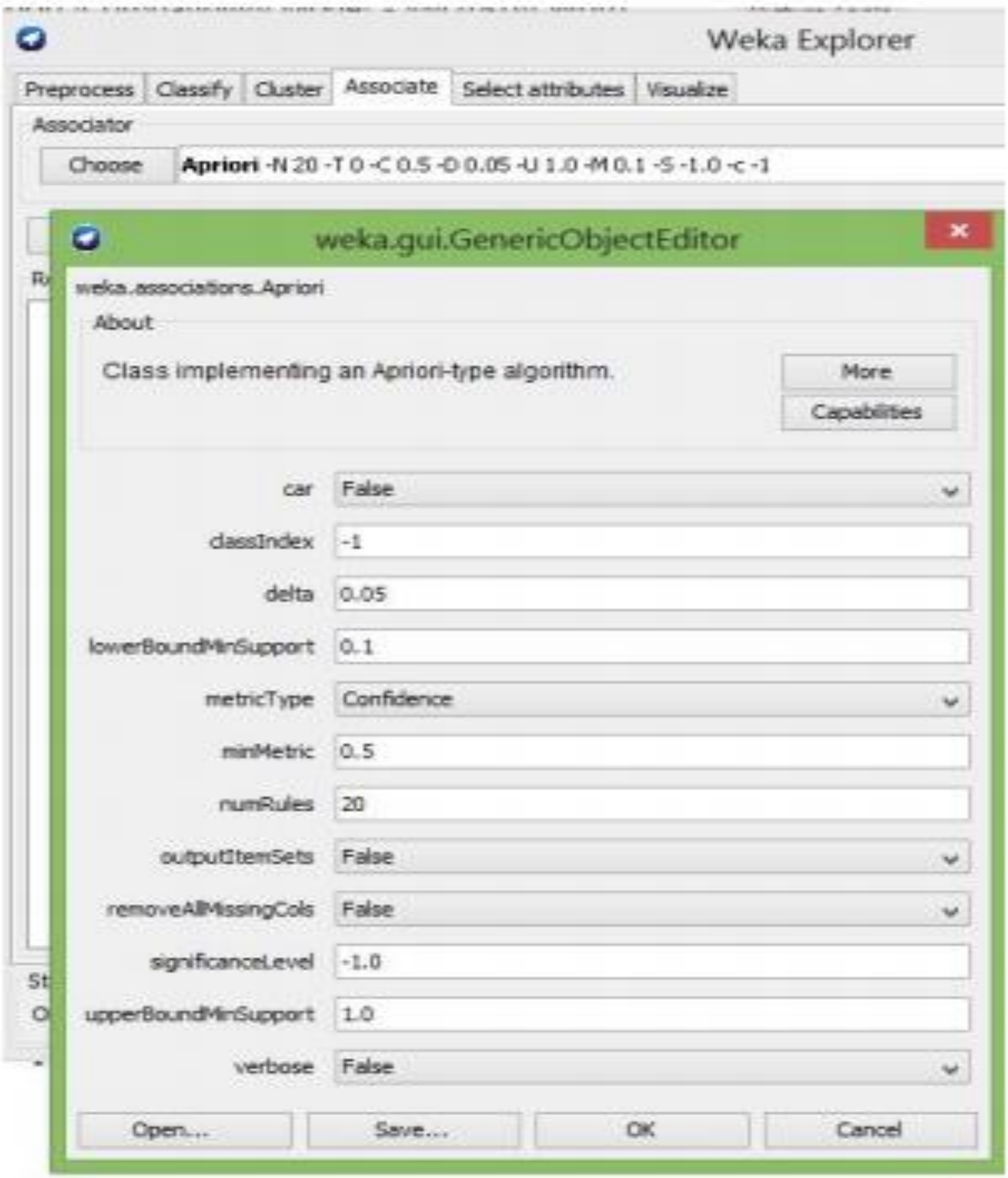
Attributes

No.		Name
1	☑	A
2	☑	B
3	☑	C
4	☑	D
5	☑	E
6	☑	F
7	☑	G
8	☑	H

Class: H (Nom)

Status
OK

Then we move to the Associate tab and we set up the configuration as shown below



Apriori -N 20 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

weka.associations.Apriori

About

Class implementing an Apriori-type algorithm.

car	False
classIndex	-1
delta	0.05
lowerBoundMinSupport	0.1
metricType	Confidence
minMetric	0.5
numRules	20
outputItemSets	False
removeAllMissingCols	False
significanceLevel	-1.0
upperBoundMinSupport	1.0
verbose	False

After the algorithm is finished, we get the following results:

```
=== Run information ===
Scheme: weka.associations.Apriori -N 20 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation: TEST_ITEM_TRANS
Instances: 15
Attributes: 8
A B C D E F G H
=== Associator model (full training set) ===Apriori =======
Minimum support: 0.5 (7 instances)
Minimum metric: 0.5 Number of cycles performed: 10
Generated sets of large itemsets:
```

```
Size of set of large itemsetsL(1): 10
Size of set of large itemsetsL(2): 12
Size of set of large itemsetsL(3): 3
Best rules found
1. E=TRUE 11 ==> H=TRUE 11 conf:(1)
2. B=TRUE 10 ==> H=TRUE 10 conf:(1)
 3. C=TRUE 10 ==> H=TRUE 10 conf:(1)
 4. A=TRUE 9 ==> H=TRUE 9 conf:(1)
 5. G=FALSE 9 ==> H=TRUE 9 conf:(1)
6. D=TRUE 8 ==> H=TRUE 8 conf:(1)
7. F=FALSE 8 ==> H=TRUE 8 conf:(1)
8. D=FALSE 7 ==> H=TRUE 7 conf:(1)
9. F=TRUE 7 ==> H=TRUE 7 conf:(1)
10. B=TRUE E=TRUE 7 ==> H=TRUE 7 conf:(1)
11. C=TRUE G=FALSE 7 ==> H=TRUE 7 conf:(1)
12. E=TRUE G=FALSE 7 ==> H=TRUE 7 conf:(1)
13. G=FALSE 9 ==> C=TRUE 7 conf:(0.78)
14. G=FALSE 9 ==> E=TRUE 7 conf:(0.78)
15. G=FALSE H=TRUE 9 ==> C=TRUE 7 conf:(0.78)
16. G=FALSE 9 ==> C=TRUE H=TRUE 7 conf:(0.78)
17. G=FALSE H=TRUE 9 ==> E=TRUE 7 conf:(0.78)
18. G=FALSE 9 ==> E=TRUE H=TRUE 7 conf:(0.78)
19. H=TRUE 15 ==> E=TRUE 11 conf:(0.73)
20. B=TRUE 10 ==> E=TRUE 7 conf:(0.7)
```

Result:

Conclusion:

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Such pre-processing is thus studied and Demonstrate performing Classification, Clustering, Association algorithm on data sets using data mining tool.

Industrial Applications: Data mining is used in various applications some of them are given below: Sales & Marketing: Data mining enables us to understand the hidden patterns inside the data that helps in planning strategy of marketing. Health Care Industry: The growth of health industry is increasing day by day. Data Mining helps to store all the data of patients those who are suffering from same type of disease. Education & Sports: In this field a vast amount of statistics data are collected for each student, teacher, subject and session. Data mining can be used by education organizations in the form of statistical analysis, pattern discovery as well as for prediction.

Questioner:

1. A goal of data mining includes which of the following?
A. To explain some observed event or condition B. To confirm that data exists
C. To analyze data for expected relationships D. To create a new data warehouse
2. Define algorithm
3. What is Apriori algorithm?
4. What is association in Data Warehousing?
5. What is pre-processing of data?
6. Explain the need for data pre-processing?
7. What kind of data can be cleaned?
8. List the data mining tool.
9. Explain R tool.
10. State the advantages of weka tool
11. Give the advantages of R tool.
12. Name areas of applications of data mining?
13. Capability of data mining is to build ___________ models.

A. retrospective. B. interrogative. C. predictive. D. imperative.